香港教育大學 The Education University of Hong Kong

Artificial Intelligence and Digital Competency Education Centre 人工智能及數碼能力教育中心

Themed-Based International Conference Series TBICS

Co-organiser: Southern University of Science and Technology 南方科技大學

The 9th International Conference on Computational Thinking and STEM Education

In the Era of AI

18 - 20 June 2025

CTE-STEM 2025

Conference Proceedings

Supporting Organisations:

CoolThink@JC 賽馬會運算思維教育

Education Bureau
The Government of the Hong Kong Special Administrative Region of the People's Republic of China

AiTLE

香港電腦教育學會 The Hong Kong Association for Computer Education

Proceedings of the 9th International Conference on Computational Thinking and STEM Education

18-20 June 2025

Hong Kong

Organized by

The Education University of Hong Kong

Co-organized by

Southern University of Science and Technology

Publication of The Education University of Hong Kong

10 Lo Ping Road, Tai Po, New Territories, Hong Kong SAR

ISSN 2664-5661

CTE-STEM 2025

Editors

Siu Cheung KONG

The Education University of Hong Kong, Hong Kong SAR

Ting Chia HSU

National Taiwan Normal University, Taiwan

Jianhua ZHAO

Southern University of Science and Technology, China

Preface

International Conference on Computational Thinking and STEM Education in the Era of AI (CTE-STEM 2025) is the ninth international conference, continuing from the success of the previous eight international Computational Thinking conferences. CTE-STEM 2025 is organized by The Education University of Hong Kong (EdUHK) and co-organized by Southern University of Science and Technology (SUSTech).

CTE-STEM 2025 is held on 18-20 June 2025. Days 1 and 2 of the conference are held at EdUHK's Tai Po Campus, while Day 3 is held at SUSTech's Campus in Shenzhen. The conference is the most remarkable event of the Programme for worldwide sharing of ideas as well as dissemination of findings and outcomes on the implementation of computational thinking and STEM education development.

The conference this year includes keynote speeches, a teacher forum and paper presentations. The Teacher Forum is held on the first day of the conference. The purpose of the Forum is to set a stage for K-12 teachers worldwide to share best practices and key challenges of implementing Computational Thinking Education (CTE) in different countries/ regions, and ultimately to facilitate CTE going global and increase involvement of K-12 teachers in the knowledge and experience exchange process.

Conference Organization:

Conference Chair

Siu Cheung KONG	The Education University of Hong Kong, Hong Kong SAR

Program Chair

Ting-Chia HSU	National Taiwan Normal University, Taiwan

Program Co-Chair

Jianhua ZHAO	Southern University of Science and Technology, China

International Program Committee Members *(Surnames in alphabetical order)*

Fahriye ALTINAY	Near East University, Northern part of Cyprus, Nicosia
Ankur BIST	Graphic Era Hill University, India
Ivica BOTICKI	University of Zagreb, Republic of Croatia
Chi-Cheng CHANG	National Taiwan Normal University, Taiwan
Ben CHANG	National Central University, Taiwan
Shao-Chen CHANG	National Taiwan Normal University, Taiwan
Yu-Shan CHANG	National Taiwan Normal University, Taiwan
Chih-Hung CHEN	National Taichung University of Education, Taiwan
Guang CHEN	Beijing Normal University, China
Jun-Ming CHEN	Soochow University, Taiwan
Ming-Puu CHEN	National Taiwan Normal University, Taiwan
Weidong CHEN	Suzhou University of Science and Technology, China
I-Chun CHEN	Chinese Culture University, Taiwan
Kwok Shing CHENG	The Education University of Hong Kong, Hong Kong SAR
Jongpil CHEON	Texas Tech University, USA
Feng-Kuang CHIANG	Shanghai Jiao Tong University, China
Yen-Lin CHIU	National Taiwan University of Science and Technology, Taiwan
Hyungshin CHOI	Chuncheon National University of Education, Korea
Chih-Ming CHU	National ILan University, Taiwan

Emerita Sara BAÑADOS SANTANA	Universidad de Concepción, Chile
Peter SEOW	Nanyang Technological University, Singapore
Rustam SHADIEV	Zhejiang University, China
Junjie SHANG	Peking University, China
Kenneth Kuen-Fung SIN	The Education University of Hong Kong, Hong Kong SAR
Yanjie SONG	The Education University of Hong Kong, Hong Kong SAR
Niwat SRISAWASDI	Khon Kaen University, Thailand
Daner SUN	The Education University of Hong Kong, Hong Kong SAR
Meng-Jung TSAI	National Taiwan Normal University, Taiwan
Pei-Shan TSAI	National Taipei University of Technology, Taiwan
Meng-Ping TSUEI	National Taipei University of Education, Taiwan
Yasuhiko TSUJI	The Open University of Japan, Japan
Jan VAHRENHOLD	Westfälische Wilhelms-Universität Münster, Germany
Haipeng WAN	Capital Normal University, China
Sally WAN	The Chinese University of Hong Kong, Hong Kong SAR
Chun-wang WEI	Far East University, Taiwan
Gary Ka Wai WONG	The University of Hong Kong, Hong Kong SAR
Sheng-Yi WU	National Tsing-Hua University, Taiwan
Wan-Hsuan YEN	National Taiwan Normal University, Taiwan
Ibrahim H. YETER	Nanyang Technological University, Singapore

Local Organizing Committee

Chair

Prof. Siu Cheung KONG	The Education University of Hong Kong, Hong Kong SAR

Member

Ho Fung Peter LAM	The Education University of Hong Kong, Hong Kong SAR
Mei Ki Vera CHAN	The Education University of Hong Kong, Hong Kong SAR
Yiling Elaine CHEN	The Education University of Hong Kong, Hong Kong SAR
Wan Yee TING	The Education University of Hong Kong, Hong Kong SAR
Luwei Vicky YE	The Education University of Hong Kong, Hong Kong SAR

Conference Theme:

Computational Thinking and STEM Education in the Era of AI

Sub-themes:

- Computational Thinking and Unplugged Activities in K-12
- Computational Thinking and Coding Education in K-12
- Computational Thinking and Subject Learning and Teaching in K-12
- Computational Thinking and Teacher Development
- Computational Thinking and IoT
- Computational Thinking Development in Higher Education
- Computational Thinking and STEM/STEAM Education
- Computational Thinking and Non-formal Learning
- Computational Thinking and Psychological Studies
- Computational Thinking and Special Education Needs
- Computational Thinking in Educational Policy
- General Submission to Computational Thinking Education
- Computational Thinking and Evaluation
- Computational Thinking and Data Science
- Computational Thinking and Artificial Intelligence Education
- Computational Thinking and its Key Elements
- Computational Thinking as Method
- STEM and Interdisciplinary Integration
- Open-Source Software and Hardware for CT and STEM Education
- Teacher Forum

The conference received a total of 47 submissions (13 full papers, 15 short papers and 19 poster papers) by 104 authors from 14 countries/regions (see Table 1).

Table 1: Distribution of Paper Submissions for CTE-STEM 2025

Country / Region	No. of Authors	Country / Region	No. of Authors
Hong Kong SAR	42	**Singapore**	3
Taiwan	23	**Spain**	3
China	8	**Israel**	2
United States	8	**Lithuania**	1
Japan	4	**Malaysia**	1
Peru	4	**Sweden**	1
India	3	**The Netherlands**	1
		Total	**104**

The International Programme Committee (IPC) is formed by 70 Members and 3 Co-chairs worldwide. Each paper with author identification anonymous was reviewed by at least three IPC Members. Related sub-theme Chairs then conducted meta-reviews and made recommendation on the acceptance of papers based on IPC Members' reviews. With the comprehensive review process, 41 accepted papers are presented (13 full papers, 14 short papers and 14 poster papers) (see Table 2) at the conference.

Table 2: Paper Presented at CTE-STEM 2025

Sub-themes	Full Paper	Short Paper	Poster Paper	Total
- Computational Thinking and Unplugged Activities in K-12	0	0	0	0
- Computational Thinking and Coding Education in K-12	0	1	0	1
- Computational Thinking and Subject Learning and Teaching in K-12	0	2	0	2
- Computational Thinking and Teacher Development	2	2	0	4
- Computational Thinking and IoT	0	0	0	0
- Computational Thinking Development in Higher Education	0	0	1	1
- Computational Thinking and STEM/STEAM Education	1	1	0	2
- Computational Thinking and Non-formal Learning	0	1	0	1
- Computational Thinking and Psychological Studies	0	1	0	1
- Computational Thinking and Special Education Needs	1	0	0	1
- Computational Thinking in Educational Policy	0	1	0	1

- General Submission to Computational Thinking Education	3	0	0	3
- Computational Thinking and Evaluation	0	1	0	1
- Computational Thinking and Data Science	0	1	0	1
- Computational Thinking and Artificial Intelligence Education	3	1	0	4
- Computational Thinking and its Key Elements	0	1	0	1
- Computational Thinking as Method	0	0	0	0
- STEM and Interdisciplinary Integration	1	1	3	5
- Open-Source Software and Hardware for CT and STEM Education	1	0	0	1
- Teacher Forum	1	0	10	11
Total	**13**	**14**	**14**	**41**

The conference comprises keynote and invited speeches by internationally renowned scholars; a teacher forum, as well as academic and poster paper presentations.

Academic and Poster Paper Presentations

There are 10 sessions of academic and poster paper presentations with 30 papers (12 full papers, 14 short papers and 4 poster papers) in the conference. Worldwide scholars present and exchange the latest research ideas and findings, which highlight the importance and pathways of computational thinking education covering K-12 education, artificial intelligence education, teacher development and STEM/STEAM education, etc.

Teacher Forum

There are 2 sessions of teacher forum paper presentations with 11 papers (1 full paper, 10 poster papers) in the conference. K-12 teachers share best practices and key challenges of implementing CTE in their countries/regions.

On behalf of the Conference Organizing Committee, we would like to express our gratitude towards all speakers as well as paper presenters for their contribution to the success of CTE-STEM 2025.

We sincerely hope everyone enjoys and gets inspired from CTE-STEM 2025.

Prof. Siu Cheung KONG

The Education University of Hong Kong, Hong Kong SAR

Conference Chair of CTE-STEM 2025

Prof. Ting-Chia HSU

National Taiwan Normal University, Taiwan

Conference Program Chair of CTE-STEM 2025

Prof. Jianhua ZHAO

Southern University of Science and Technology, China

Conference Program Co-Chair of CTE-STEM 2025

Table of Contents

Full Paper

Short Paper

<u>Poster Paper</u>

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

A Preliminary Approach to Quantitative Evaluation of Modified Problem-Posing for Problem Structure Understanding in Computational Thinking

Masanori FUKUI[1*], Ryohei MIRADERA[2], Yuji SASAKI[3], Tsukasa HIRASHIMA[4]
[1]Iwate Prefectural University, Japan
[2]Keimei Gakuin Senior High School, Japan
[3,4]Hiroshima University, Japan
fukui_m@iwate-pu.ac.jp, runnerskg@gmail.com, y.sasaki@keio.jp, tsukasa@lel.hiroshima-u.ac.jp

Abstract: *Understanding and quantitatively assessing problem structure is a key challenge in computational thinking (CT) education. This study proposes and preliminarily examines "modified problem-posing" as a means to transform and quantify how learners adapt an existing problem—here, the Collatz conjecture—to deepen structural understanding. We tested 21 high school and 37 university students. Results showed that high school students tended to focus on a single category change, while university students often changed multiple categories simultaneously. These findings illustrate differences in complexity and indicate that a deeper, quantitative analysis of problem transformations is feasible. This work offers new possibilities for systematically evaluating problem-posing activities, contributing to more effective CT-oriented instruction.*

Keywords: *Problem-posing, Understanding of the problem structure, Computational thinking, Classroom, Quantitative Evaluation*

1. Introduction

In the 21st century, computational thinking (CT) is increasingly recognized as a core problem-solving skill alongside creativity (Wing, 2006). CT is the process of organizing and expressing problems in a form that computers can solve (Wing, 2006). In addition, this also involves dividing the problems and solving them. CT involves formulating problems so that computers (or algorithmic processes) can address them, encompassing decomposition, abstraction, and identifying commonalities. This emphasis on problem structure has been noted as crucial for effective learning and problem solving (Lee et al., 2016). It is essential to conduct further research on CT development and increase the CT of subjects, as CT is a problem-solving skill using computer technology, including AI (Wing, 2006). To improve problem-solving skills using AI and other technologies, it is important not only to promote education to improve problem-solving skills, which has been conducted thus far, but also to improve the ability to understand problem structures and consider how to solve problems in future education.

The authors proposed "modified problem posing" to improve problem structure comprehension skills as a framework for enhancing problem structure comprehension skills based on CT. (Fukui, 2024). Modified problem-posing is a method based on learning to create problems by oneself. It is a framework for promoting problem structure understanding by transforming a given problem. This framework is an important basic skill for CT and creativity; modified problem-posing aims to develop the skills needed in future society (Fukui & Sasaki, 2022). The framework is useful not only for developing the ability to create problems but also for problem-solving and shows examples of applying modified problem-posing to programming and mathematics education (Fukui & Kuroda et al., 2024; Fukui & Sasaki, 2022). In the future, examining their usefulness in practice and relevance to creativity and CT will be necessary. However, only a simplified method for evaluating modified problem-posing has been proposed; an evaluation has not yet been conducted.

The purpose of this study is to examine evaluation methods for modified problem presentation, which has been proposed as a way to improve problem structure understanding. We conducted a pilot study with high school and university students and present a method for systematically tabulating the number and combinations of transformations. We also discuss the possibility of enabling more quantitative evaluation of problem structure understanding, which has not been rigorously pursued in previous studies.

2. Research Design

2.1. About Modified Problem-posing

Understanding problem structure is expected to improve problem solving and basic creativity skills (Hunter et al., 2008). Problem-posing learning is a type of learning that improves understanding of problem structure. Problem-posing learning involves creating problems by oneself (Mishra, 2014). It is useful for improving creative thinking, critical thinking, learning motivation (Kaur & Rosli, 2021), and understanding the problem structure (Silver, 1994). Learning support systems that use written arithmetic problems as subjects for problem-posing have also been developed (Hirashima et al., 2007), and their usefulness has been widely acknowledged.

However, it is easy to imagine that some students can create many problems while others cannot when engaged in problem-posing. It has also been pointed out that questions/problems created by students who have never created questions before lack diversity and that more diversity is needed (Mestre, 2002). Therefore, it is important to consider improving learners' understanding of the problem structure by learning composition questions and problem-posing.

To solve this problem, the author proposed a modified problem-posing method enabling novice learners to create their problems. Modified problem-posing is an activity that creates new problems by transforming and improving a given problem. It is a framework that promotes an understanding of the problem structure and enhances the basic skills of creativity (Fukui et al., 2019). Modified problem-posing encourages learners to transform a given problem systematically. By tracking which elements are changed, we can quantify the scope and complexity of each learner's transformations. This potentially yields new insights into how deeply learners grasp a problem's structure (Polya, 1990; Fukui & Sasaki, 2022). Besides, modified problem posing is related to CT (Fukui & Sasaki, 2022). The relationship between modified problem-posing and CT is presented in Table 1.

Table 1. Relationship between computational thinking and modified problem-posing (Fukui and Sasaki, 2022)

Concepts of computational thinking	Examples of activity
Decomposition	Find the changeable part of the problem
Generalization	Find common point for changeable parts of the problem
Abstraction	Find common point that can be used in other problems
Algorithmic thinking	Accurately represent the flow of the problem

2.2. Example of The Materials in Modified Problem-posing

In this study, the Collatz problem is used as the subject of the modified problem-posing. We used the Collatz conjecture as our target problem (3n+1 problem). It starts with a natural number n, dividing by 2 if even, and multiplying by 3 then adding 1 if odd, repeating until reaching 1. Figure 1 shows the flowchart (Fukui & Sasaki, 2022). It can be broken into six categories (I, D, Op, T, R & P). We instructed participants to alter one or more categories (e.g., branching conditions, operations, terminal conditions), either singly or in combination. Examples of transformations are shown in Table 2. For example, you could change "the given natural number n" to a natural number n with a range, change it to a real number r, or change the part "if it's even, do process A, if it's odd, do process B" to a branch that does something other than even or odd, or change processes A and B. It is also possible to combine these in complex ways.

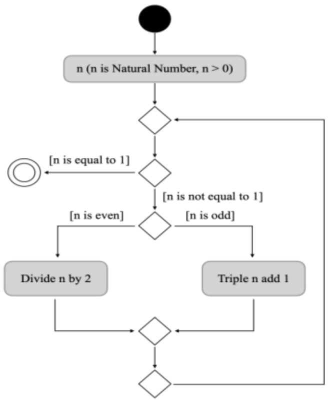


Figure 1. The Flowchart of Collatz Conjecture (Fukui & Sasaki, 2022)

Table 2. Definition of categories

Name of Category	Meaning
I	Initial condition
D	Decision condition
Op	Operation/calculation
T	Terminal
R	Repeat times
P	The number of Players

Table 3. Examples of modifications

Focus point	Changing part	Example of variants
Value n in natural number	Initial condition	To restrict the range of n Change n to a real number
Branching by even or odd numbers	Decision condition	Branching by multiples of three or other
Processing part after branching, divide n by two, or triple n and add one	Operation/calculation	Processing part after branching, divide n by three, or double n
End of play if it reaches 1	Terminal condition	End of play if it reaches n ($n \neq 1$)
Repeat until finished	Repeat condition	Repeat three times

3. Research Method

3.1. Survey Targets and Survey Procedure

In September 2018, 21 senior high school students (private school) and 37 second-year university students (majored in information science) were taught the Collatz conjecture, then asked to create variants within 15 minutes. The percentage of valid responses was 100%. Participation was voluntary, and responses were anonymized.

3.2. Survey Item

Students were presented the Collatz conjecture and asked to create as many variants as possible. They could optionally invalidate the problem. We aimed to observe how categories changed.

3.3. Analysis Procedure

We classified each variant by which categories were changed (Table 2), then compared high school vs. university students. For learners who created multiple variant problems, the number of categories modified, and the number of categories adopted were used. Sample size was small, so detailed sub-analyses are limited.

4. Results and Discussions

The differences between the original questions and the revised questions were evaluated and classified based on the results of the experiment. The results are shown in Table 4. As a result, university students were most likely to change two categories, followed by three. High school students were most likely to change one category, followed by two. No high school students changed four or six categories.

In addition, changing only one category accounted for 29.3% of the total, and this was particularly common among high school students. Changing two categories accounted for 32.8% of the total, and this trend was also seen among university students and high school students. Changing three categories accounted for 19.0% of the total, and this was particularly common among university students. The tendency to change the number of categories to 4 or 6 was only seen in university students. The tendency to change the number of categories to 5 was seen in 8.6% of all cases, and this was seen almost equally in university students and high school students.

While university students tended to change multiple categories or multiple parts of the problem, high school students were most likely to change one category and least likely to change multiple categories. In particular, there was a difference between high school students and university students when changing 3 or more categories.

Table 4. Results of modifications

Number of Categories	Total (n = 58)	University (n = 37)	High School (n = 21)
1	17(29.3%)	7(18.9%)	10(47.6%)
2	19(32.8%)	11(29.7%)	8(38.1%)
3	11(19.0%)	10(27.0%)	1(4.8%)
4	5(8.6%)	5(13.5%)	0(0.0%)
5	5(8.6%)	3(8.1%)	2(9.5%)
6	1(1.7%)	1(2.7%)	0(0.0%)

Table 5. Results of modifications

Category	Total (n = 58)	University (n = 37)	High School (n = 21)
Op	17(29.3%)	7(18.9%)	10(47.6%)
C,Op	15(25.9%)	8(21.6%)	7(33.3%)
Op,T	4(6.9%)	3(8.1%)	1(4.8%)
C,Op,I	2(3.4%)	1(2.7%)	1(4.8%)
C,Op,T	2(3.4%)	2(5.4%)	0(0.0%)
Op,I,T	2(3.4%)	2(5.4%)	0(0.0%)
I,P,T	4(6.9%)	4(10.8%)	0(0.0%)
P,R,T	1(1.7%)	1(2.7%)	0(0.0%)
C,Op,I,P	1(1.7%)	1(2.7%)	0(0.0%)
Op,I,P,T	2(3.4%)	2(5.4%)	0(0.0%)
Op,P,R,T	2(3.4%)	2(5.4%)	0(0.0%)
C, I,Op,P,T	2(3.4%)	2(5.4%)	0(0.0%)
C,I,Op,P,R	1(1.7%)	0(0.0%)	1(4.8%)
C,Op,P,R,T	1(1.7%)	0(0.0%)	1(4.8%)
I,Op,P,R,T	1(1.7%)	1(2.7%)	0(0.0%)
C,I,Op,P,R,T	1(1.7%)	1(2.7%)	0(0.0%)

Table 5 shows the results for each category that was changed or improved. For example, C+Op indicates that both C and Op were corrected at the same time. Op was the most common category to be changed, especially among high school students. C+Op was the most common category to be changed for multiple categories, and only C+Op and Op+T were seen among high school students.

One difference between university students and high school students is that high school students tend to change one category at a time, whereas university students tend to change two or three categories at the same time. This is thought to be because university students are used to solving complex problems and are not averse to changing multiple elements at the same time. Ogilvie (2009) points out that when faced with complex open-ended tasks such as solving physics problems, university students tend to adopt a wider range of more considered strategies. Furthermore, the motivation for complex changes among university students may come from academic fields that require the integration of various knowledge domains and application in practical contexts.

Furthermore, high school students tended to make the most frequent changes in one category, particularly in the manipulation (Op) category. This may be since high school students prefer simpler problem-solving strategies and tend to focus on changes in one category. Wüstenberg et al. (2014) point out that high school students tend to rely on single-step solutions and have difficulty tackling complex, multi-layered problems. This may be related to the transformation of problem-solving. In addition, university students are highly motivated to solve complex problems and tend to tackle more multifaceted tasks (Balta et al, 2016). This indicates that the frequency of multi-category changes might reflect a more complex or holistic approach to problem structure. Moreover, the presence or absence of certain combinations (e.g., C+Op+T) might offer a quantifiable metric for deeper structural engagement. Future research could exploit these combination patterns as an index of problem-structure comprehension—an approach not yet extensively explored in existing problem-posing literature.

5. Conclusion

This study proposed a preliminary evaluation method for modified problem-posing using the Collatz conjecture. Our findings reveal that high school students mainly changed a single category, while university students altered multiple categories simultaneously. These differences potentially arise from educational background and motivation.

Importantly, we have also shown that by categorizing and counting each transformation (*e.g.*, Op, C+Op, Op+T, etc.), there is potential for a quantitative measure of how deeply learners engage with the problem's structure. Such a quantitative approach to problem transformation is relatively unexplored, opening avenues for deeper analysis of CT skill development.

However, limitations remain regarding sample size and the lack of direct motivation measures. As we advance, larger-scale experiments and more detailed correlation with external measures (e.g., problem-solving ability, creativity) will be crucial. Nonetheless, we believe that modified problem-posing, combined with a structured evaluation framework, can serve as an innovative method to both teach and assess CT-based problem comprehension in ways previous research has rarely accomplished. Furthermore, some participants created multiple variants of the problem. For instance, they might have changed only the C category in their first version, then changed both C and Op in the second, and later changed C and D in the third. Because multiple submissions came from the same individual, additional caution is needed when interpreting these results. In future research, more careful analysis of multiple variants per participant will be necessary to accurately capture the complexity of their modifications.

Furthermore, this study did not directly measure the participants' motivation and interest in problem transformation; thus, assessing their influence on the results was impossible. Therefore, collecting data on the participants' motivations and interests in future research is important. Since there are various patterns of change in each category, these patterns need to be analyzed in detail to clarify the difficulty of the transformations and improvements. Considering the above limitations, future research should be conducted on a larger scale and in greater detail to collect more accurate data for designing educational programs. It is also necessary to develop educational programs based on previous studies focusing on the relationship between CT and external measures (Fukui, Sasaki & Hirashima, 2022; Fukui, Xiang et al, 2024). These issues need to be addressed in future studies.

References

Balta, N., Mason, A., & Singh, C. (2016). Surveying Turkish High School and University Students' Attitudes and Approaches to Physics Problem Solving. *Physical Review Physics Education Research*, *12*, 010129. https://doi.org/10.1103/PhysRevPhysEducRes.12.010129.

Boden, M. (1998). Creativity and artificial intelligence. *Artificial Intelligence*, *103*, 347-356. https://doi.org/10.1016/S0004-3702(98)00055-1

Collard, P., & Looney, J. (2014). Nurturing creativity in education. *European Journal of Education*, *49*, 348–364. https://doi.org/10.1111/EJED.12090

Fukui, M. (2024). A proposal of a method to promote creativity based on computational thinking using modified problem-posing. *DINAMIKA Jurnal Ilmiah Pendidikan Dasar*, *16*(1), 41–47. https://doi.org/10.30595/dinamika.v16i1.21467

Fukui, M., Kuroda, M., Chew, P., & Sasaki, Y. (2024). Proposal for developing learning materials focusing on modified problem-posing using a probability game to promote understanding of the problem structure in basic engineering mathematics. *International Journal of Applied Physics and Mathematics*, *14*(1), 35–44. http://dx.doi.org/10.17706/ijapm.2024.14.1.35-44

Fukui, M., Sasaki, Y., & Hirashima, T. (2022). An exploratory study of the relationship between computational thinking and creative attitudes among university students. *The 6th APSCE International Conference on Computational Thinking and STEM Education 2022 (CTE-STEM 2022)*.

Fukui, M., & Sasaki, Y. (2022). Proposal of a small-c creativity cultivation method with modified problem-posing for use in computer studies education. *Proceedings of 11th DATTArc-ICTE-TENZ-ITEEA*. [in print]

Fukui, M., Xiang, L., Sano, Y., Yanuarto, W. N., Anggoro, S., Chew, P., Ong, E. T., & Ng, K. T. (2024). An exploratory study of the relationship between fixed/growth mindset and computational thinking among university students. *The 8th APSCE International Conference on Computational Thinking and STEM Education 2024 (CTE-STEM 2024)*.

Hirashima, T., Yokoyama, T., Okamoto, M., & Takeuchi, A. (2007). Learning by problem-posing as sentence-integration and experimental use. *International Conference on Artificial Intelligence in Education*, 254–261.

Hunter, S., Bedell-Avers, K., Hunsicker, C., Mumford, M., & Ligon, G. (2008). Applying multiple knowledge structures in creative thought: Effects on idea generation and problem-solving. *Creativity Research Journal*, *20*, 137–154. https://doi.org/10.1080/10400410802088779.

Kaur, A., & Rosli, R. (2021). Problem posing in mathematics education research: A systematic review. *International Journal of Academic Research in Progressive Education and Development*, *10,* 438–456. https://doi.org/10.6007/IJARPED/V10-I1/8641

Lee, H. S., Betts, S., & Anderson, J. R. (2016). Embellishing problem-solving examples with deep structure information facilitates transfer. *Journal of Experimental Education*, *85*(2), 309–333. https://doi.org/10.1080/00220973.2016.1180277

Mestre, J. P. (2002). Probing adults' conceptual understanding and transfer of learning via problem posing. *Journal of Applied Developmental Psychology*, *23*, 9–50. http://dx.doi.org/10.1016/S0193-3973(01)00101-0

Mishra, S. (2014). Developing students' problem-posing skills. *Proceedings on the Tenth Annual Conference on International Computing Education Research (ICER '14)*, 163–164. https://doi.org/10.1145/2632320.2632333

Ogilvie, C. (2009). Changes in Students' Problem-Solving Strategies in a Course that Includes Context-Rich, Multifaceted Problems. *Physical Review Special Topics - Physics Education Research*, *5*, 020102. https://doi.org/10.1103/PHYSREVSTPER.5.020102.

Polya, G. (1990). *How to solve it: A new aspect of mathematical method*. Penguin.

Scott, S., & Koch, D. (2010). Teaching university-level technology students via the learning preferences and problem-solving approach. *The Journal of Technology Studies*, *36*(1), 16–23.

Silver, E. A. (1994). On mathematical problem posing. For the Learning of Mathematics, 14(1), 19–28.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wüstenberg, S., Greiff, S., Molnár, G., & Funke, J. (2014). Cross-national gender differences in complex problem solving and their determinants. *Learning and Individual Differences*, *29*, 18–29. https://doi.org/10.1016/j.lindif.2013.10.006

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

A Study on the Application of Digital Learning Partners in the Digital Technology Course

Siang-Chun HSIEH [1], Yi-Wen LIAO[2*]
[1,] National Kaohsiung Commercial High School, Kaohsiung, Taiwan
[2] National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan
ciao.scshow@gmail.com, pinkwen923@gmail.com

Abstract: *Digital learning partners are an essential auxiliary tool in modern digital learning, enhancing learners' efficiency and self-directed learning abilities. This study aims to explore the application of digital learning partners (such as the Adaptive Learning Platform) in an Introduction to Digital Technology course. By integrating digital teaching materials and designing learning activities, the study seeks to improve students' learning outcomes. The research focuses on incorporating four types of self-directed learning methods into the course: individual self-learning (self-regulation), teacher-guided learning (external regulation), collaborative learning within groups (co-regulation), and inter-group peer learning (social shared regulation). Furthermore, this study evaluates the impact of digital learning partners on students' academic performance. The findings are expected to enhance the interactivity and adaptability of digital learning environments and provide valuable insights for future digital course design and AI-based education applications.*

Keywords: Self-Directed Learning, Digital Learning Partner, Artificial Intelligence, Learning Performance

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

應用數位學習夥伴於數位科技概論課程之研究

謝享淳[1]，廖奕雯[2*]
[1]國立高雄高商資料處理科，台灣
[2]國立高雄科技大學智慧商務系，台灣
ciao.scshow@gmail.com，pinkwen923@gmail.com

【摘要】 數位學習夥伴是當前數位學習中一項重要的輔助工具，能夠提升學習者的學習效率與自主學習能力。本研究旨在探討如何運用數位學習夥伴(如因材網) 於數位科技概論課程，並透過數位教材錄製及學習活動設計，提升學生的學習成效。研究關注如何在課程中導入自主學習的四種學習方式，包括學生自學(自我調節)、教師導學(他者調節)、組內共學(共同調節)、組間互學(社群共享節點)，並評估數位學習夥伴對學生學習表現的影響。本研究預期可提升數位學習環境的互動性與適應性，並提供未來數位課程設計與 AI 教育應用的參考。

【關鍵字】 自主學習；數位學習夥伴；人工智慧；學習表現

1. 前言

隨著科技和教育創新的不斷發展，如今的學習者可以從教師、同儕、教育科技以及最近的生成人工智慧(如 ChatGPT)等代理處獲得各種支援。學術界對人機協作和混合智慧學習的興趣激增。混合智慧的概念仍處於起步階段，學習者如何從與人工智慧、人類專家和智慧學習系統等各種代理的共生關係中受益，也是目前學者投入研究的方向 (Fan et al., 2024)。

隨著人工智慧技術的快速發展，人工智慧(如 ChatGPT、因材網 AI 夥伴等)在教育領域的應用愈加廣泛。數位科技概論課程涵蓋多元知識，且科技變遷迅速，傳統教學模式難以即時更新。因此，應用數位學習夥伴，能夠提供即時回饋、個性化學習建議，並提升學生的自主學習能力。本研究關注如何透過相關技術提升學生學習效果，並探討其在自主學習模式中的應用。研究目的為：

1. 探討數位學習夥伴應用於數位科技概論課程的影響。
2. 設計與評估基於數位學習夥伴輔助的自主學習模式 (自學、導學、共學、互學)。
3. 分析數位學習夥伴對學生學習成效的影響，並評估其適用性。
4. 提供在數位學習環境中的最佳實踐方案，為未來數位課程設計提供參考。

2. 文獻探討

2.1. 自主學習 (self-regulated learning, SRL)

自主學習(self-regulated learning, SRL) 是指學習者在學習過程中，主動設定學習目標，並透過計畫、監控和調整自己的認知、動機和行為，以達成這些目標的過程。這一概念強調學習者的自主性和自我導向，認為有效的學習者能夠自我管理、自我監控和自我調節，以適應不同的學習需求和環境 (Zimmerman, 2002; Pintrich, 2004; Schunk & Zimmerman, 2008)。

Miedijensky et al., (2025) 研究探討了在線和創新學習環境中設計、實施和評估自我調節學習體驗的方法，強調了學生在自主學習中的挑戰和機遇。Greenquist-Marlett et al., (2025) 研究調查了小學教師在課堂中實施自我調節學習策略的情況，以及這些策略與教師自我效能感之間的關聯。Karlen et al., (2024) 研究強調了教師的元認知技能作為自我調節學習者，對學生 SRL 能力的理解和培養具有重要影響。Luo & Zhou (2024) 研究評估了在過去五年中，自我調節學習策略在高等教育混合式學習環境中的有效性，結果顯示 SRL 策略對學習成果有積極

影響。這些文獻深入探討了自我調節學習的理論基礎、評估方法以及在教育實踐中的應用，對理解和促進學生的自主學習具有重要參考價值。

2.2. 因材網

「因材網」是由臺灣教育部於 2017 年 3 月正式上線的教師適性教學輔助平台，旨在提升學生學習效果和教師教學效能。該平台涵蓋數學、自然科學和國語文等學科，適用於一至九年級學生。其主要功能包括：

(1) 知識結構學習：協助學生了解各學科的知識架構，建立完整的學習脈絡。
(2) 智慧適性診斷：透過精細且快速的診斷工具，識別學生的學習弱點，提供個人化的學習路徑。
(3) 互動式學習：提供教學影片、互動式元件和動態評量等多媒體資源，增進學生的學習興趣與自主學習能力。
(4) 21 世紀核心素養：培養學生的關鍵能力，適應未來社會的挑戰。

截至 2019 年 6 月，已有超過 39 萬名國中小師生使用因材網，顯示其在教育界引發廣泛關注與應用。 此外，因材網還提供操作手冊、增能研習影片等資源，協助教師熟悉平台功能，提升教學品質。 總之，因材網透過適性教學與多元資源，致力於促進學生的學習成效，並提升教師的教學效能。

3. 研究方法

3.1. 教案內容

本研究授課單元/主題為：「連接網際網路的方式：有線」，課程的教學目標為：(1)了解有線連接網際網路的方式與原理；(2) 認識常見的有線網路連接設備及功能；(3) 應用有線網路連接技術於其生活實務。

課程的教學方式為：教師授課、觀看教學影片/老師講解、課堂討論。教學相關資源為：(1) 教學簡報；(2) 因材網「連接網際網路的方式：有線」教學影片；(3) 教學表單：自主學習規劃單、WSQ 學習單、分組紀錄單、自主學習反思單。教學應用的知識得採用「WSQ 學習單」，知識應用為「錄影分享法、主題式討論」教育部因材網，學習畫面如圖 1 所示。

圖1 因材網學習平台

3.2. 課程設計

課程設計包含數位教材錄製、AI 學習夥伴及自主學習模式設計等，自主學習設計四學四個階段說明，如表 1 所示。

- **數位教材錄製**：錄製數位科技概論課程內容，並整合 AI 互動機制。
- **AI 學習夥伴應用**：使用因材網 AI 作為數位學習夥伴，提供個性化學習建議與即時反饋。
- **自主學習模式設計**：
  - **自學(自我調節)**：學生透過 AI 夥伴進行自主學習。

- **導學(他者調節)**：教師利用 AI 進行個別輔導與進度監控。
- **共學(共同調節)**：學生透過 AI 進行小組討論與協作學習。
- **互學(社群共享節點)**：不同小組透過 AI 共享學習成果與知識。

課堂採用的學習方式為科技輔助學習模式，如圖 2 所示。課程學習單如圖 3 所示。

表1 課程設計四學

課程階段	教學活動	教材與使用之科技	時間
教師導學 **【教師導學】** **【自主反思】**	**《直接引導學習》** 學習反思活動—學生依據自主學習規劃單自訂學習目標。 一、引導階段 (一)介紹有線連接網際網路的方式 1.簡述網際網路的連接方式分為有線及無線 2.說明本次課程為學習有線連接網際網路的方式 (二)教師提問 題目：家中的電腦設備是如何連接網際網路？	自主學習規劃單 教學簡報	5 分鐘
【自學】 **【組內共學】**	二、Watch 階段 (一)觀看教學影片 有線連接網際網路的基本方式，涵蓋「網路設備的功能」及「連接方式」。 **《WSQ 學習單》** (二)討論與補充 1.根據影片內容，讓學生能夠了解日常生活中，如何讓電腦設備能夠透過有線的方式來進行連接網際網路。 2.學生藉由填寫 WSQ 學習單，進行課堂討論及引導。 **《主題式討論》** 三、Summarize 階段 (一)分組討論 1.學員分組討論影片與講解的內容。 2.每組總結關鍵點並寫在白紙上。	因材網教學影片 WSQ 學習單 分組紀錄表	5 分鐘 5 分鐘 10 分鐘 10 分鐘
【組間互學】	(二)小組分享 1.每組選一位代表分享小組討論之總結。 2.教師補充並強調重點。 **《直接引導學習》** 四、Question 階段 (一)學生提問 引導學生思考在實際生活中對於有線連接網際網路的應用。	分組紀錄表	10 分鐘
【教師導學】	(二)教師解答	自主學習	5 分鐘

課程階段	教學活動	教材與使用之科技	時間
	1.回答學生的問題，並提供進一步的概念說明。 2.確認學生是否明確了解課程內容。 五、總結與作業 (一)課程總結 回顧本次課程的主要內容，強調的重要性。 (二)布置作業 請學生填寫自主學習反思單來反思自我的學習狀況。	反思單	

圖2 科技輔助自主學習模式

表 2. WSQ 學習單

觀察及記錄(W)	這堂課請觀看以下單元主題的影片，完成請打勾。 ☑ 四村網「連接網際網路的方式：有線」影片
	1. 已完成觀看單元主題之「電話線撥接上網」影片部分，完成請打勾。 ☑ 電話線撥接上網是透過家用的電話線路撥號連上網際網路 ☑ 電話線路若被用來傳送資料，則無法撥打或接聽電話
	2. 已完成觀看單元主題之「ADSL 上網」影片部分，完成請打勾。 ☑ ADSL 上網是採用「電話語音訊號」及「網路傳輸訊號」分離的技術上網 ☑ 以「下載速度／上傳速度」來表示頻寬速度
	3. 已完成觀看單元主題之「纜線數據機上網」影片部分，完成請打勾。 ☑ 纜線數據機上網是利用有線電視業者的纜線系統 ☑ 若共用同一條纜線的用戶數增多時，則連線速度會變慢
	4. 已完成觀看單元主題之「光纖上網」影片部分，完成請打勾。 ☑ 光纖上網的線材為光纖電纜 ☑ 常見的光纖上網方式：光纖到府(FTTH)、光纖到大樓(FTTB)、光纖到路側(FTTC)
	5. 已完成觀看單元主題之「專線上網」影片部分，完成請打勾。 ☑ 由電信業者提供的固定線路讓使用者能隨時連線 ☑ 常用的線路有 T1、T2、T3、T4 等規格
總結(S)	針對這些單元主題的課程內容你都理解了嗎？有理解的部分，請打勾。 ☑ 電話線撥接上網 ☑ ADSL 上網 ☑ 纜線數據機上網 ☑ 光纖上網 ☑ 專線上網 請摘要紀錄你學會的以上這些課程： 都是有線連接到網路的方式
提問(Q)	提問(Q)回想你的學習過程，有發現不了解的地方嗎？請在以下列出你不了解的地方。 無

表 4. 自主學習反思單

題目	學習反思問題	回答選項
1	這個單元的測驗，我實際獲得了幾分？	80 分
2	在這堂課的學習努力程度，我給自己打幾分？	1. 我對於自己的學習成果，給予 100 分。 (2.) 我對於自己的學習成果，給予 90 分。 3. 我對於自己的學習成果，給予 80 分。 4. 我對於自己的學習成果，給予 70 分。
3	完成這堂課的學習後，你認為自己是否有學好這堂課呢？	(1.) 有，我覺得自己學得很好。 2. 一半一半，我覺得有些概念我還不是很懂。 3. 沒有，我覺得我還沒有把這堂課的知識學得很好。
4	請回想你自己當初設定的目標，你是否確實依照自己目標執行呢？	1. 有，我當初很有把握，我現在確實完成。 (2.) 一半一半，我還需要依據自己的程度調整目標。 3. 沒有，目標跟我的表現差異很大。
5	你實際採用哪些方法進行預習或複習呢？	1. 我下課會用 5-10 分鐘時間進行學習。 2. 我會再加多觀看老師提供的影片來學習。 (3.) 我會上網蒐集跟有線連接網際網路相關的影片進行學習。 4. 我會寫習題，並詢問老師或者班上較厲害的同學。 5. 我會寫習題，並詢問學校以外的專業人士。
6	根據你自己的學習成果以及學習方法，你覺得有哪些需要修改進的地方？	(開放性問題) 無

圖3 課程學習單

4. 研究結果

以某高中職開設之「數位科技概論」課程學生為研究對象，收集 31 名受試者，分別為男學生 12 位及女學生 19 位。本研究採用量化數據分析 AI 數位學習夥伴對學習成效的影響。

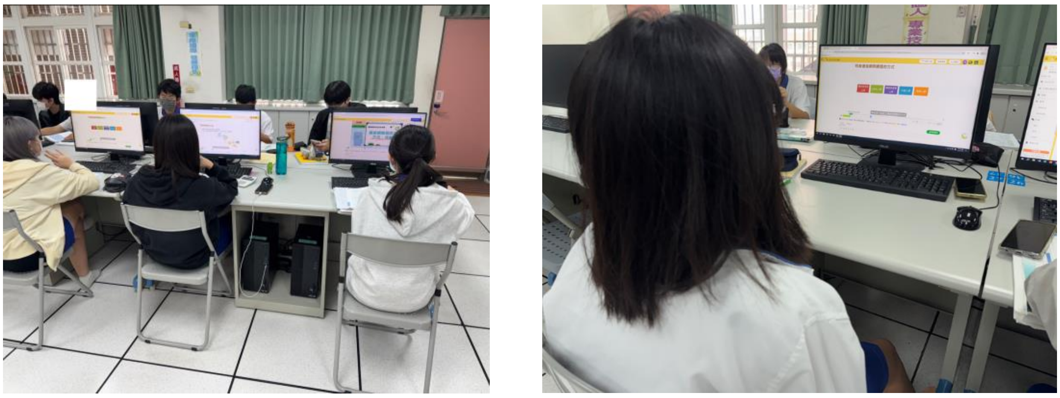


圖4 教學活動現場

表2 學習成效分析

	N	平均數	標準差	變異數	T 檢定
前測分數	31	53.55	20.42	416.99	1.038 (t)
M	12	58.33	22.09		0.308 (sig.)
F	19	50.53	19.29		
後測分數	31	82.80	18.29	334.63	-1.77 (t)
M	12	75.83	19.29		0.087(sig.)
F	19	87.37	16.61		

1. **前後測評估**：進行學習前後測試，分析 AI 輔助學習對學生學習成效的影響。
2. **學習行為分析**：透過學習平台記錄學生學習時間、互動頻率、提問次數等數據。
3. **問卷調查與訪談**：收集學生對 AI 學習夥伴的滿意度與使用經驗。

課程實施前後，分別針對學生對課程內容的成效檢驗，分析結果如表 2 所示。前測平均分數為 53 分，後測平均分數為 82 分，結果顯示透過教材課程錄製及因材網學習，可有效提升學生的學習成效。另外，性別在後測分數有顯著差異，女生在後測學習成效表現比男生優異。

5. 討論與結論

本研究透過實施結果發現，可有效提升學生學習成效，並透過影片及因材網學習方式增強學生的自主學習能力，相關研究貢獻說明如下：

1. **提升學習成效**：透過 AI 輔助的自主學習模式，提升學生的理解能力與學習成就。
2. **增強自主學習能力**：幫助學生建立有效的學習策略，提高學習自律性。
3. **最佳化 AI 在教育領域的應用**：提供 AI 輔助數位學習的實證研究，作為未來教育技術應用的參考。
4. **跨領域應用價值**：本研究成果可延伸至其他學科領域，促進 AI 技術在教育領域的多元應用。

本研究探討生成式 AI 在數位科技概論課程中的應用，並評估其對學生學習成效與自主學習能力的影響。預計研究結果將顯示 AI 學習夥伴能夠有效促進學生的學習參與度，並提升數位學習環境的互動性。未來研究可擴展至其他學科領域，或探討不同 AI 模型對學習成效的影響，進一步驗證 AI 在教育科技領域的長期價值。

参考文獻

Fan, Y., Tang, L., Le, H., Shen, K., Tan, S., Zhao, Y., ... & Gašević, D. (2024). Beware of metacognitive laziness: Effects of generative artificial intelligence on learning motivation, processes, and performance. *British Journal of Educational Technology*.

Greenquist-Marlett, S., Bol, L., & Hill, C. (2025, January). Teachers' perceptions of their self-regulated learning practices in elementary school classrooms. In *Frontiers in Education* (Vol. 9, p. 1464350). Frontiers Media SA.

Karlen, Y., Hertel, S., Grob, U., Jud, J., & Hirt, C. N. (2024). Teachers matter: linking teachers and students' self-regulated learning. *Research Papers in Education*, 1-28.

Luo, R. Z., & Zhou, Y. L. (2024). The effectiveness of self-regulated learning strategies in higher education blended learning: A five years systematic review. *Journal of Computer Assisted Learning*, *40*(6), 3005-3029.

Miedijensky, S., Sasson, I., & Glick, D. (2025). Designing, implementing and evaluating self-regulated learning experiences in online and innovative learning environments. *Frontiers in Education*, 10, 1546434. https://doi.org/10.3389/feduc.2025.1546434

Pintrich, P. R. (2004). A conceptual framework for assessing motivation and self-regulated learning in college students. *Educational Psychology Review*, 16, 385-407. https://doi.org/10.1007/s10648-004-0006-x

Schunk, D. H., & Zimmerman, B. J. (Eds.). (2008). *Motivation and self-regulated learning: Theory, research, and applications*. Lawrence Erlbaum Associates.

Zimmerman, B. J. (2002). Becoming a self-regulated learner: An overview. *Theory into Practice*, 41(2), 64-70. https://doi.org/10.1207/s15430421tip4102_2

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Bridging the AI Literacy Gap:

A Constructivist, No-Code AI Curriculum for Secondary Students

Phylliscia CHEW*, Da LI*
NUS High School of Math and Science, Singapore
nhscsp@nus.edu.sg, nhsld@nus.edu.sg

Abstract: *As artificial intelligence (AI) becomes an integral part of modern life, fostering AI literacy among secondary students is imperative. Existing AI education resources vary in accessibility and depth, with some presenting technical barriers or lacking coherence, impeding students' holistic understanding of fundamental concepts. This paper introduces a structured, no-code AI curriculum that leverages computational thinking and constructivist pedagogy to promote accessibility without compromising rigor. Through hands-on, visual-first learning tools, students explore AI ethics, machine learning (ML), and generative AI without requiring prior programming experience. Classroom implementation results demonstrate significant improvements in AI literacy, critical thinking, and ethical reasoning, underscoring the curriculum's scalability and adaptability as a model for secondary AI education.*

Keywords: AI literacy, Computational thinking, constructivist learning, No-code AI education, Ethical AI,

1. Introduction

As artificial intelligence (AI) technologies rapidly evolve, secondary students increasingly interact with AI-powered tools such as ChatGPT without possessing the necessary conceptual and ethical frameworks to understand their inner workings, biases, and limitations. Since 2016, NUS High School has incorporated Computational Thinking into its Secondary 1 curriculum, originally focusing on problem-solving, programming principles, and data skills. In 2024, the course was restructured to introduce structured AI-focused units.

While several free AI education resources—such as IBM SkillsBuild AI Courses, Microsoft AI for Beginners, and Intel AI for Youth—offer valuable content, many presuppose a degree of coding proficiency, which may create barriers for novices. Meanwhile, no-code initiatives like aiEDU, AI4ALL, Experience AI, and MIT Day of AI have significantly broadened access, though they often prioritize terminology or introductory activities over sustained engagement with machine learning (ML) processes.

To address this issue, we developed a scalable AI curriculum rooted in constructivist learning theory and computational thinking. The curriculum is distinguished by three innovations: (1) hands-on engagement using visual, no-code platforms to demystify ML pipeline; (2) integration of technical concepts with ethical reasoning; and (3) a strong commitment to accessibility without sacrificing conceptual rigor. By moving beyond surface memorization toward meaningful, critical exploration, our curriculum empowers students to build authentic AI literacy.

By adopting a constructivist, no-code approach, this curriculum empowers students to move beyond surface-level definitions and develop meaningful, critical, and ethical AI literacy skills.

2. Curriculum Design

To promote genuine understanding rather than surface memorization, the curriculum was intentionally designed to avoid overwhelming Grade 7 students with technical terms such as "supervised learning" or "reinforcement learning," which are often abstract and disconnected from their everyday experiences. Without firsthand experience in coding and implementing those advanced concepts, students may struggle to fully appreciate their significance. Therefore, the curriculum adopts a gradual, scaffolded learning progression: it begins with accessible, hands-on activities to introduce foundational AI concepts, transitions into no-code exploration of the machine learning pipeline and builds toward the real-world applications and limitations of generative AI. The learning journey culminates in a research-based exploration of AI ethics. This structured approach ensures students develop both technical fluency and ethical reasoning, preparing them for deeper engagement with AI technologies.

Our curriculum is strategically structured around five core design principles below, each supported by concrete instructional strategies and student-centered activities.

2.1. Democratizing AI Education

Many traditional AI courses require significant programming and mathematical expertise, creating barriers for learners without prior experience. Our curriculum eliminates these obstacles through a no-code, visual-first approach, making AI concepts accessible to all students, regardless of prior coding experience or socioeconomic background.

In Unit 1, students engage with Google Teachable Machine, providing an intuitive, hands-on introduction to machine learning. This interactive experience allows them to train simple models without writing code, fostering curiosity and foundational AI literacy.

Building on these fundamentals, Unit 2 introduces hands-on machine learning, where students transition from basic experimentation to structured AI workflows. Rather than coding ML models from scratch, they utilize no-code ML pipelines in Google Colab, where they upload datasets, fine-tune parameters, and analyze model outputs, all without requiring programming skills. To maintain accessibility while ensuring conceptual depth, students evaluate model performance using high-level indicators like loss vs epochs diagram, the value of accuracy, Mean Absolute Error, confusion matrices, and regression scatter plots (Figure 1, 2). These visual tools allow students to interpret ML results meaningfully without delving into complex mathematical derivations. To further enhance understanding, an animated visual representation of the Neural Network Training Process (Figure 3) illustrates the key process in an engaging, digestible format.

To explore AI hallucinations, students critically examine inconsistencies in AI-generated responses. For example, they investigate how ChatGPT provides conflicting explanations for the 'P' in 'ChatGPT,' with responses ranging from 'procedural' to 'predictive' to having no specific meaning. This activity reinforces the importance of source verification and critical evaluation. A recorded conversation with an earlier ChatGPT version, showcasing its hallucinations, provides a controlled, replicable learning experience.

2.2. Balancing Accessibility and Technical Depth

While accessibility remains a cornerstone of our curriculum, intellectual depth is equally prioritized. Students engage in hands-on AI workflows that mirror real-world machine learning applications, progressing through all five ML pipeline stages: Define the Problem, Data Collection & Preprocessing, Model Training, Model Evaluation, and Iterative Refinement. Through guided experimentation, they manipulate key parameters—such as the number of hidden layers in a neural network—to assess their impact on model accuracy and performance.

To illustrate the concept of overfitting, students first train models with varying hidden layer counts in Google Colab, observing first-hand that increasing model size does not always enhance accuracy. This is reinforced through cognitive anchoring, a learning strategy where new information is linked to familiar concepts to improve understanding. In this case, overfitting is likened to a multi-level video game player who memorizes every detail of a single level but struggles when faced with new ones—just as an overfitted model fails to generalize beyond its training data. This dual approach fosters both conceptual clarity and practical comprehension.

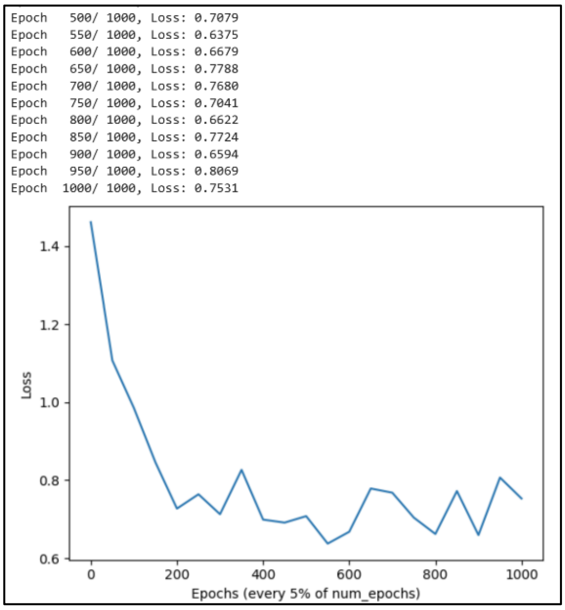


Figure 1. The Loss vs Epochs graph demonstrates how the model improves over multiple training rounds. This visualization presents numerical feedback in a simplified manner, making learning progress observable without complex mathematical explanations

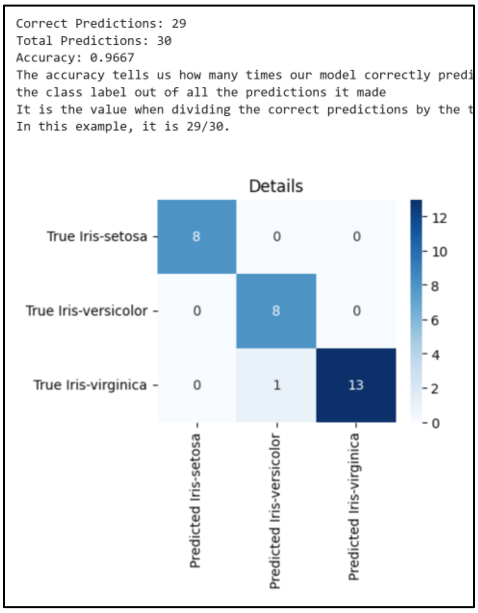


Figure 2. Accuracy with contextual explanations introduces evaluation concepts to students. Correct and incorrect predictions are shown numerically, while technical terms like 'confusion matrix' are omitted to enhance accessibility

Large Language Models (LLMs) are similarly demystified through intuitive analogies that make abstract concepts more accessible. For example, our work compares the process of text generation to a Shakespearean playwright selecting words— each word chosen must fit the sentence's structure and context. An LLM achieves this using statistical methods rather than predefined rules. However, like an author injecting creativity, LLMs introduce an element of randomness, allowing for more dynamic and varied text generation. By framing LLM behavior through familiar concepts, students grasp both the structured and probabilistic nature of AI without needing advanced technical knowledge.

Additionally, the curriculum integrates emerging AI research, such as findings on model collapse when training on synthetic data, underscoring the necessity of high-quality datasets in AI model development.

2.3. Hands-On and Personalized Learning

Beyond theoretical understanding, active participation enhances learning outcomes. The curriculum leverages hands-on exploration and personalized engagement to deepen students' conceptual grasp and foster critical thinking. In the No-Code ML Pipeline work, students work with self-selected datasets, allowing them to apply AI concepts to personally meaningful topics, enhancing engagement and understanding.

A custom-built AI playground allows students to explore how *top_K* settings influence AI-generated responses, even when using the same prompt. By adjusting these settings, students observe responses ranging from predictable and balanced to highly imaginative—or, at extreme values, nonsensical gibberish or non-English characters (Figure 4). This hands-on activity provides a concrete demonstration of the probabilistic nature of AI, reinforcing the importance of critical evaluation when interpreting AI-generated content. Students explore AI ethics by conducting independent research on topics of their choice, using a structured, iterative approach. They begin with a preliminary exploration using an LLM to gain insights, then conduct independent online investigations to identify and analyze real-world case studies. Finally, they return to LLM for synthesis and deeper discussion. Their findings culminate in written reflections on AI's societal implications and a concrete action plan for ethical AI use appropriate for their age. To support this process, a worked case study on AI-driven vehicles provides step-by-step guidance, complete with teacher's commentary to scaffold learning at each stage.

By engaging with these real-world challenges through hands-on exploration and personalized learning, students develop critical thinking, problem-solving, and ethical reasoning—key competencies in computational thinking education.

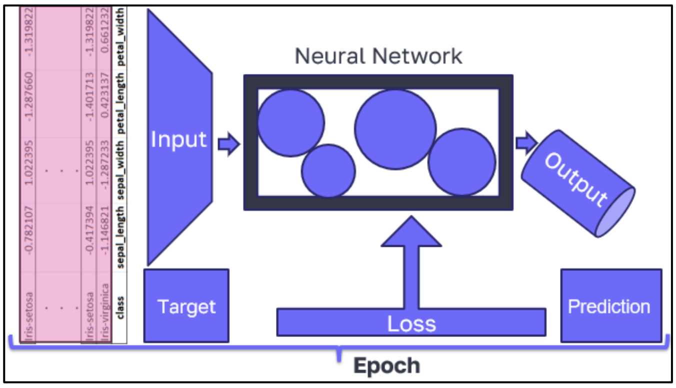


Figure 3. Visual illustration of neural network training. The neural network is represented as a box, with dynamic circle sizes indicating parameter changes over training steps. The diagram visually introduces key ML concepts and processes while avoiding complex mathematics.

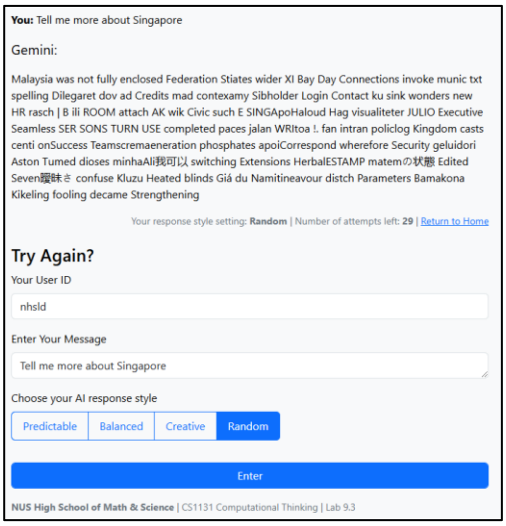


Figure 4. Interface of the custom-built AI playground. In this example, the 'Random' setting produces gibberish with non-English characters, demonstrating the impact of extreme randomness.

2.4. Ethical AI Education: A Structured and Progressive Approach

As students develop technical competencies, it is imperative to integrate ethical considerations throughout their learning journey. AI ethics is embedded as a core pillar of our curriculum, ensuring that responsible AI use is not an afterthought but an integral component of AI literacy. Ethical considerations are interwoven with technical instruction, allowing students to engage with bias, misinformation, and fairness through hands-on experimentation rather than abstract discussion. While Unit 4 (AI Ethics & Responsible AI Use) provides a structured space for deeper inquiry, ethical engagement is embedded throughout the curriculum from the very beginning:

- Unit 1: Introduction to AI & ML – Students explore AI's pattern recognition through visual demos, prompting critical discussions on bias in datasets.
- Unit 2: No-Code ML Pipeline – By adjusting parameters and analyzing model outcomes through visual tools, students examine fairness in algorithmic decision-making.
- Unit 3: Generative AI & Prompt Engineering – Students interact with AI-generated content, investigating hallucinations, misinformation, data security and the reliability of LLMs.
- Unit 4: CS Ethics & Responsible AI Use – With a strong conceptual foundation, students critically assess real-world AI applications, evaluating their societal impact and ethical challenges.

By structuring ethics as an evolving discussion across technical units, students cultivate a deep, experience-driven understanding of responsible AI use. This approach cultivates both AI proficiency and ethical discernment, preparing students to navigate complex AI applications in academic and real-world contexts.

Unit 4, the most critical unit of the curriculum, equips students with a structured framework for analyzing AI ethics. It addresses key concerns such as fairness, intellectual property, data privacy, and AI's impact on employment. Through a case-based learning approach, students examine real-world failures—such as the 2024 CrowdStrike outage—to understand AI vulnerabilities and their broader societal implications. Discussions on dark patterns in digital design expose manipulative engagement strategies, fostering awareness of ethical pitfalls in AI-driven applications.

A pivotal component of this unit is the structured discourse on AI-generated content and plagiarism, where students critically examine the boundaries of responsible AI adoption in academic settings. This ensures they develop ethically grounded AI literacy, reinforcing accountability, critical evaluation, and institutional integrity.

2.5. Scalable, Open-Source, and Teacher-Friendly Design

Designed for broad adoption, the curriculum is built on free, web-based platforms, for example, Google Teachable Machine, Google Colab, GitHub, ChatGPT and Google Gemini, eliminating financial barriers and ensuring accessibility across diverse educational settings. Structured teaching resources equip educators—regardless of AI background—with the necessary tools to deliver effective instruction. This modular, adaptable framework positions the curriculum as a scalable, future-ready model for AI education in secondary schools.

3. Implementation & Impact

Integrated as the final component of the 2024 Grade 7 Computational Thinking Course, the AI curriculum spanned nine instructional hours, balancing direct instruction with hands-on experimentation. Table 1 summarizes the key components, assessments, and personalized learning strategies. Designed for universal accessibility, it required no prior AI or coding experience. Teachers provided real-time support, allowing weaker students to receive guidance while stronger students engaged in self-directed exploration, including custom datasets, iterative ML training, and independent research.

Table 1. Key Component of Each Unit

Unit	Hours	Key Contents	Examples of Assessment	Element of Personalized Learning
1	2	Hierarchical understanding of AI and ML. AI’s real-world applications	Google Teachable Machine	Self-taken images
2	4	Simplified the Neural Network Training Process, supported with visual aids.	No-Code ML Pipeline Using Google Colab	Self-sourced dataset
3	3	Mechanics and limitations of LLMs, Comparison & combination of the use cases of LLMs and online search	Explore Randomness Setting of LLMs	Personalized prompts
4	2	Clearly defining rules with contextual examples for ethical AI use	AI Ethics Mini-Research Task	Self-selected topic

Post-course surveys demonstrated significant gains in AI literacy, engagement, and ethical awareness:

- 95.5% of students agreed it enhanced their AI understanding.
- 91.0% reported greater competency in generative AI tools and ethical discussions.
- 88.7% affirmed that the curriculum’s objectives were met.

Student reflections highlighted strong engagement, with many expressing confidences in AI discussions and a desire for deeper exploration of both technical and ethical dimensions. These findings affirm the curriculum’s effectiveness and scalability as a model for AI education, demonstrating its potential for broader adoption in secondary-level computational thinking programs.

4. Future Directions & Expansion

Building on the successes and insights gained from this implementation, we now look toward future curriculum enhancements and broader scalability. Now a permanent part of the Grade 7 curriculum, our AI program is updated annually to reflect emerging advancements, enhance engagement, and ensure relevance. As AI continues to evolve rapidly, future iterations of the curriculum will introduce topics like Retrieval-Augmented Generation (RAG) and AI-assisted learning to deepen students' understanding. The program is set to expand to 12 hours, incorporating other innovations such as comparisons between Small and Large Language Models and interdisciplinary applications.

Comprehensive teacher training and structured resources facilitate seamless implementation across diverse educational contexts, while open-access platforms ensure equitable AI literacy at scale.

5. Conclusion

This study responds to the pressing need to equip secondary students with authentic AI literacy, going beyond vocabulary-building or pen-and-paper activities common in existing no-code AI initiatives. Our constructivist, no-code curriculum is original in its holistic approach: students not only engage deeply with the machine learning pipeline through hands-on, visual tools but also develop critical ethical reasoning across all stages of learning.

Grounded in both constructivist learning theory and computational thinking, the curriculum enables students without coding experience to develop meaningful AI literacy, critical thinking, and ethical discernment. Implementation results demonstrate significant gains in student understanding and engagement, confirming the curriculum's scalability and adaptability for secondary education.

As AI continues to reshape society, this curriculum offers a practical, innovative model for preparing future generations as informed and responsible AI practitioners.

Acknowledgements

We sincerely thank our school leaders for their support and guidance, and our colleagues for their contributions to the development and implementation of the AI curriculum. Their insights, expertise, and commitment were instrumental in shaping this initiative and advancing our educational goals.

References

Akgun, S., & Greenhow, C. (2022). Artificial intelligence in education: Addressing ethical challenges in K-12 settings. *AI & Ethics, 2*, 431–440. https://doi.org/10.1007/s43681-021-00096-7

AI4ALL. (n.d.). *AI4ALL Homepage.* AI4ALL. Retrieved April 28, 2025, from https://ai-4-all.org/

Chew, P., & Li, D. (2021). Designing a computational thinking curriculum for everyone with a differentiated and gamified approach. *Proceedings of the Fifth APSCE International Conference on Computational Thinking and STEM Education Teachers Forum,* 32-25. https://cte-stem2021.nie.edu.sg/assets/docs/CTE-STEM_Teachers-Forum_Compiled-Proceedings.pdf

Jiahong, S., Zhong, Y., & Ng, D. T. K. (2022). A meta-review of literature on educational approaches for teaching AI at the K-12 levels in the Asia-Pacific region. *Computers and Education: Artificial Intelligence*, *3*, 100065. https://doi.org/10.1016/j.caeai.2022.100065

Matti Tedre. 2022. Computational Thinking 2.0. *Proceedings of the 17th Workshop in Primary and Secondary Computing Education (WiPSCE '22)*. 1–2. https://doi.org/10.1145/3556787.3556788

MIT RAISE. (n.d.). *Day of AI. Day of AI*. Retrieved April 28, 2025, from https://dayofai.org/

Raspberry Pi Foundation & Google DeepMind. (n.d.). *Experience AI. Experience AI*. Retrieved April 28, 2025, from https://experience-ai.org/en

Shumailov, I., Shumaylov, Z., & Zhao, Y. et al. (2024). AI models collapse when trained on recursively generated data. *Nature*, *631*, 755–759. https://doi.org/10.1038/s41586-024-07566-y

The AI Education Project (aiEDU). (n.d.). *The AI Education Project.* aiEDU. Retrieved April 28, 2025, from https://www.aiedu.org/

Computational Thinking and Artificial Intelligence Training Program for Students with Intellectual Disability: A Path to Inclusion

Natalia GALBÁN-OJER[1*], María ZAPATA-CÁCERES[2], Nardie FANCHAMPS[3], Estefanía MARTIN-BARROSO[2]
[1]Fundación Ciganda Ferrer-Colegio E.E. El Molino, Spain
[2]Universidad Rey Juan Carlos, Spain
[3]Open University, The Netherlands
ngalban@centroselmolino.org, maria.zapata@urjc.es, nardie.fanchamps@ou.nl, estefania.martin@urjc.es

Abstract: *Computational Thinking (CT) facilitates knowledge acquisition and skill development, particularly in problem-solving and understanding information processes based on computer science principles. However, its potential to address socio-pedagogical and psychological needs for learners with developmental disabilities remains underexplored. Integrating CT with Artificial Intelligence (AI) could enable individuals with intellectual disabilities to navigate an increasingly digital society inclusively. This study examines how CT, applied to Spanish students with intellectual disability (ID), can reduce the digital divide and enhance employability while exploring AI's role in this process. A qualitative research program analyzed observable behaviors following CT and AI training. The results indicate that integrating CT with AI positively impacts learning in the following areas: (a) digitalization and CT, (b) block-based programming, and (c) AI training systems using machine learning. Findings suggest that CT and AI integration improves problem comprehension, logical structuring, pattern recognition for enhanced solutions, differentiation between relevant and irrelevant information, and the ability to develop clear, step-by-step strategies. While the study provides positive indications, further research is needed to deepen the understanding of CT and AI's role in socio-pedagogical and psychological support for students with intellectual disabilities.*

Keywords: Computational Thinking, Artificial Intelligence, Intellectual Disabilities, Inclusive Education, Employability.

1. Introduction

With the rise of artificial intelligence (AI) and the digital transformation of society, it is necessary to implement training programs that support the acquisition of knowledge and skills that prepare students to successfully face contemporary challenges in a critical and resilient way (Shute et al., 2017). From inclusive education, it is essential to guarantee the access of students with intellectual disability to this type of training programs, in order to improve their social and labor inclusion and minimize the existing digital divide (UNESCO, 2019). The employment rate of this group according to the latest report from the Spanish National Institute of Statistics (2023) is 21.9%, a figure that highlights the need for training programs to improve their social and labor insertion. In addition, women with intellectual disability suffer double discrimination in access to employment due to gender and disability stereotypes (Sánchez & Pérez, 2020).

Intellectual disability (ID) is characterized by significant limitations in intellectual functioning and adaptive behavior, which pose challenges in the acquisition of conceptual, social, and practical skills before the age of 18 (Matson et al., 2009). This individual variability requires an adequate personalization of the teaching process (Palacios-García, 2024) under an adaptive learning approach that strengthens their potentialities (Selau et al., 2022).

UNESCO (2019) describes Computational Thinking (CT) as a key competence of the 21st century for learning digital technologies and artificial intelligence (AI) in the classroom. CT favors the deployment of cognitive skills for problem solving through the expression of solutions such as automated sequences or algorithms that can be executed by both computer agents and/or without them (Wing, 2006). This process is divided into 1) Understanding the problem; 2) Decomposition into easily solvable subtasks; 3) Representation of the process and interpretation of the data; 4) Implementation of a sequence of steps or algorithms; and 5) Debugging and validation of the solution (Shute et al. 2017). CT constitutes a new essential literacy for STEM learning at all educational stages, impacting the development of higher cognitive skills through the improvement of reasoning and decision-making processes in an orderly, systematic, sequenced and logical way with application to the resolution of everyday problems (Burke et al., 2016). In this sense, CT is not reduced to the transmission of computer concepts but facilitates the understanding of natural and social phenomena and the explanation and interpretation of the world as a composite of information processes (Denning & Tedre, 2019).

CT can be worked on firstly, through disconnected activities, which do not require the use of a computational agent; and secondly, through connected activities, which employ digital tools and programming (Fanchamps et al., 2021; Saxena & Shanahan, 2020), including AI (Olmo-Muñoz et al., 2020). This adaptive approach to CT is aligned with the use of cognitive scaffolds as facilitators of a problem-solving process (Ojeda, 2011). It is precisely this scaffolding that allows students with ID to self-regulate their own learning through metacognitive processes. (Chaves, Rodríguez, & Ramírez, 2006).

In response to the demand for STEM training for students with ID and with the aim of facilitating their inclusion in digital and AI-related contexts, this study raises the following main research question: Does the combined implementation of CT and AI activities have a positive impact on students with ID?

To answer this main research question, the following sub-questions have been asked: (1) What specific cognitive and social benefits do CT and AI activities provide for students with ID?; (2) How do scaffolding-based instructional strategies affect student´s self-perception on their abilities in these activities?; (3) What challenges and facilitators influence the effective implementation of CT and AI activities in inclusive educational settings?

The research has been conducted under the following hypotheses: (1) Combined CT and AI activities enhance cognitive and social skills in students with ID; (2) Instructional strategies that incorporate scaffolding improve student´s self-perception of their abilities in these activities, (3) The effective implementation of these activities for students with ID depends on teacher training, accessibility of materials and institutional support.

2. Method

To explore our research questions, we gathered qualitative data through direct teachers' observations focusing on CT skills and an online self-questionnaire about digital and job abilities. The method chosen for this qualitative research was the temporary case study following the guidelines of this type of research according to Galeano (2012): 1) This program focuses on a specific event or social aspect without generalizing, and 2) describing the situation in detail and evaluating it to support future educational interventions.

2.1. CT and AI training program

This program is offered at a special school in Spain. It is available to individuals with ID who have completed the Special Vocational Training Cycle at twenty-two years old. These studies vary according to the legislation of each region. After this cycle, there is a paucity of specialized continuing education for people with ID. Consequently, CT and AI training program is aimed at people with ID over twenty-two years of age who wish to broaden their professional skills in the field of training in Machine Learning systems. The goal is, under the right to lifelong learning, the development of digital skills applied to tagging and data revision in AI systems, to promote their socio-occupational inclusion.

This program has a duration of ten months from September to June, five hours per day and five days per week. Eight people are enrolled during 2024-25 course, two of them are women. The training modules and their link to the development of CT skills (Selby, 2017; Román-González et al., 2017, Brennan & Resnick, 2012) are outlined below in table 1. They are also related to the five areas of competence established by the European Framework of Digital Competences for Citizenship (Voukari et al., 2022).

Table 1. Modules, CT skills and Digital Competences.

<table>
<tr><th>Modules/activities</th><th>CT Skills</th><th>Digital Competences</th></tr>
<tr><td>TM1 – Digitalization
- Monitoring instructions in a set order to complete a task.
- Digitize text in Microsoft Word.
- Entering data in Microsoft Excel.
- Review of tasks before submission.
- Make corrections and improve the work.</td><td>Identification of digital processes into parts.
Pattern recognition: Identification of trends in the use of digital tools.
Abstraction: Use of digital models to represent information.
Algorithmic thinking: Tracking sequential digital processes and flows.
Evaluation and debugging: Verification of digital tools and troubleshooting.</td><td rowspan="3">- Information and data literacy
- Online communication and collaboration
- Digital content creation
- Digital security
- Troubleshooting</td></tr>
<tr><td>TM2 – Block Programming
- Completing the CODE challenges. (pre-readers course)
- Use of sequences and loops.
- Making a programming game</td><td>Decomposition: Division of problems into blocks.
Patter recognition: Detection of repetitive structures.
Abstraction: Creation of simplified block algorithms.
Algorithmic thinking: Design of logic sequences.
Evaluation and debugging: Testing and correcting code.</td></tr>
<tr><td>TM3 – AI training systems. Machine Learning
- Preprocessing text, audio and image data.
- Tagging of text, audio and image data.
- Data review and validation.</td><td>Decomposition: Separation of relevant data in models.
Patter recognition: Pattern identification in data sets.
Abstraction: Building mathematical models from data.
Algorithmic thinking: Development of steps in model training.
Evaluation and debugging: Model optimization.</td></tr>
</table>

2.2. Participants

This case study has been carried out with eight students, comprising six male and two female individuals. They all have a mild-moderate range of intellectual disability between 33% and 65% according to international criteria such as those of the Diagnostic and Statistical Manual of Mental Disorders (DSM-5) (Gómez & García, 2008). All subjects have access to a computer and demonstrate literacy competence. Furthermore, they exhibit adequate oral and written comprehension and expression, employing accessible patterns of simple language.

2.3. Assessment instruments and procedure

Two qualitative data collection instruments have been designed in this case study. Firstly, a direct observation register (Table 2) has been developed for the four teachers participating in the program. Secondly, each student has been provided with an online self-assessment questionnaire. The subsequent section presents the direct observation register, which is predicated on CT skills, observable behavior and frequency.

Table 2. Direct observation register.

CT Skills	Observed behavior	Frequency
Understanding problem & decomposition	Explains the problem and divide the problem into logical and organized parts.	Always/ Sometimes/ Never
Pattern recognition	Identifies patterns and uses them to improve solutions.	Always/ Sometimes/ Never
Abstraction	Distinguishes relevant from irrelevant information.	Always/ Sometimes/ Never
Algorithmic thinking	Design a clear and efficient step-by-step strategy.	Always/ Sometimes/ Never
Evaluation and debugging	Identifies errors and improves solutions.	Always/ Sometimes/ Never

This register was completed midway through the course, as part of the formative assessment. The record was also completed again at the end of the course for the final assessment. Additionally, each student must completed an online self-questionnaire about digital and job skills. It consists of twenty-two questions tied to each training module, demonstrating high internal consistency (Cronbach's $\alpha = 0.90$). with anonymous responses gathered during weekly 30-minute tutoring sessions after prior explanation of the Likert scale from 1 to 5. The questionnaire was reviewed by CT experts and special education experts to ensure content and face validity, and all four teachers took part in a brief calibration workshop before data collection to harmonize use of the observation register and standardize scoring criteria.

3. Results and data-analysis

In order to be able to answer the main research questions by means of the formulated sub-questions and hypotheses to be investigated, qualitative data collected from the observation register and online self-assessments were analyzed.

3.1. Digital and job skills self-questionnaire

The data presented in figure 1 show the average response results for each question in the questionnaire for general trends. The highest average value (4.75) corresponds to the degree of satisfaction expressed by the students towards the training plan, followed by the activities related to the programming challenges (4.5) on the CODE platform. In contrast, the lowest average value (3.38) relates to activities associated with the oral communication of ideas.

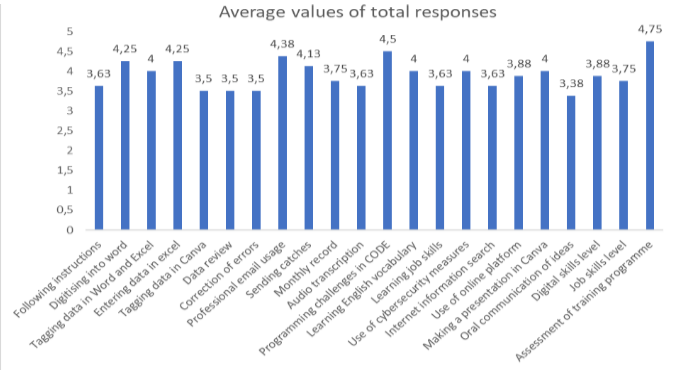


Figure 1. Bar chart of average responses.

The descriptive analysis of the questions most closely linked to CT and AI skills is shown in Table 3.

Table 3. Questions linked to CT and AI skills.

	Following instructions					Tagging data in Word, Excel					Tagging data in Canva					Data review					Correction of errors					Programming challenges in CODE				
CT skills	Decomposition					Pattern recognition and abstraction					Pattern recognition and abstraction					Evaluation and debugging					Evaluation and debugging					Algorithmic thinking				
MODE	4					4					4					3 and 4					4					4 and 5				
Scale rating	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Frequency	1	0	1	5	1	0	0	2	4	2	1	0	2	4	1	0	1	3	3	1	0	1	2	5	0	0	0	0	4	4
Percentage	13	0	13	63	13	0	0	25	50	25	13	0	25	50	13	0	13	38	38	13	0	13	25	63	0	0	0	0	50	50

The responses tend to cluster around values 3, 4, and 5, suggesting a medium-high self-perception of students' competence in CT and AI activities within the training program.

3.2. Direct observation register

The questionnaire was completed for each student by the four teachers of the training plan. Teachers highlight the high levels of motivation, sustained concentration, self-regulation and resilience of students in the face of new tasks or corrections favoring the development of adaptive skills. They underline the importance of adapting activities to each student's individual needs. In addition, teachers point out that breaking down tasks and systematizing the problem-solving process creates predictable and successful environments to foster the autonomy and self-esteem of students with ID.

The results indicated a high variability of frequencies (Always, Sometimes, Never) between students and observed performance according to PC skills, in line with the great variability of competencies shown by students with ID based on the type of task and supports provided. In the skill Decomposition, the scale ‘Sometimes’ is the most frequent among students with 37.5%. In the identification of patterns there is variability with 25% in ‘Always’, 25% in ‘Sometimes’ and 25% between ‘Sometimes-Never’. In Abstraction the most common scale is ‘Sometimes’ with 50%. In Algorithmic thinking 37.5% corresponds to the scale ‘Sometimes’ and another 37.5% to ‘Never’. And finally, in Evaluation, the most repeated scale with 62.5% is ‘Sometimes’.

4. Conclusions

The analysis reveals that integrating CT and AI activities positively impacts students with ID by enhancing cognitive problem-solving skills such as decomposition, pattern recognition, error checking, and solution validation. However, the high variability in CT skills indicates that while some students made notable progress, others needed additional support, emphasizing the need for differentiated instruction (Selau et al., 2022).

Regarding instructional strategies, scaffolding proved effective in improving students’ self-perception. Self-reported data indicate a medium-high level of confidence in digital and job-related skills, highlighting the importance of structured guidance. This aligns with research showing that scaffolding supports gradual skill development in inclusive settings (Pea, 2004).

Key factors affecting CT and AI implementation in inclusive classrooms include teacher training, accessibility of materials, and institutional support. The variability in CT skills assessed by teachers underscores the necessity for adaptive methodologies that cater to diverse learning needs (Grover & Pea, 2013).

Overall, while CT and AI activities enhance learning for students with ID, their effectiveness depends on instructional approaches and contextual factors. Future longitudinal studies could explore the long-term benefits of integrating these tools into inclusive education.

In conclusion, structured teaching strategies and supportive learning environments are crucial for fostering both skill acquisition and self-confidence in students with ID. By refining instructional methods and ensuring sufficient resources, educators can leverage CT and AI to promote inclusive and equitable learning opportunities, ultimately improving students’ social and labor inclusion.

Acknowledgments and compliance with ethical standards

Thanks to students and teachers involved who agreed to participate in this learning experience. This work is co-funded by the Erasmus+ project CoTEDI, which is also co-financed by the European Union under the call-key action 2023-1-NL01-KA220-SCH-000152037 – OID E10207981 and supported by MOVETIA Switzerland. This research was approved by the Research Ethics Committee of Universidad Rey Juan Carlos (N. 291120234412023).

References

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 Annual Meeting of the American Educational Research Association (AERA).

Burke, Q., Byrne, W.I. and Kafai, Y.B., 2016. Computational participation: Understanding coding as an extension of literacy instruction. *Journal of Adolescent & Adult Literacy*, *59*(4), 371-375. https://doi.org/10.1002/jaal.496

Chaves, A. P. V., Rodríguez, S. L., & Ramírez, M. J. R. (2006). Metacognition as a strategy to detect cognitive processes in children with intellectual disabilities. *Revista Electrónica Educare, 9*(2), 169-185. http://www.redalyc.org/articulo.oa?id=194119272008.

Denning, P. J., & Tedre, M. (2019). Computational thinking: A disciplinary perspective. *Informatics in Education, 20*(3), 361. https://www.ceeol.com/search/article-detail?id=979293

Fanchamps, N., Slangen, L., Hennissen, P. & Specht, M. (2021) The influence of SRA programming on algorithmic thinking and self-efficacy using Lego robotics in two types of instruction. *Int J Technol Des Educ 31*, 203–222 (2021). https://doi.org/10.1007/s10798-019-09559-9

Galeano, M. E. (2016). *Designing Projects in Qualitative Research.* Fondo Editorial Universidad EAFIT.

Gómez, M., & García, J. (2008). Diagnosis and Classification in Intellectual Disability. *Intervención Psicosocial, 17*(2), 157-165. https://doi.org/10.5093/inpsic.2008.v17n2.4

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Instituto Nacional de Estadística. (2023). INE. https://www.ine.es/jaxi/Datos.htm?tpx=49183

Ojeda, R. (2011). Cognitive Scaffolding Strategies in an Online Collaborative Learning Environment and Problem-Solving in an Authentic Task: Testing a Model. *HETS Online Journal*, *2*(1), 4-46. https://doi.org/10.55420/2693.9193.v2.n1.92

Olmo-Muñoz, J., Cózar-Gutiérrez, R. & González-Calero, J.A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education, 150*. https://doi.org/10.1016/j.compedu.2020.103832

Palacios-García, T. (2024). Curricular Adaptations and Their Importance for Students with Special Educational Needs. *Revista Cienciamatria 10*(18), 313–326. https://doi.org/10.35381/cm.v10i18.1273

Pea, R. D. (2004). The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. *Journal of the Learning Sciences, 13*(3), 423–451. https://doi.org/10.1207/s15327809jls1303_6

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, *72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Sánchez, L., & Pérez, A. (2020). Women with Intellectual Disabilities: Additional Challenges for Their Labor Inclusion. *Revista de Educación Inclusiva, 22*(4), 256-278. https://doi.org/10.3011/redu2020.22.4.5

Saxena, A., & Shanahan, L. (2020). Designing unplugged and plugged activities to cultivate computational thinking: An exploratory study in early childhood education. *Asia-Pacific Journal of Teacher Education, 48*(3), 251–269. https://doi.org/10.1007/s40299-019-00478-w

Selau, T., Hallberg, S. C. M., Schütz, A., & Bandeira, D. (2022). Systematic review and meta-analysis of interventions in adaptive behavior for children and adolescents with intellectual disabilities. *Journal of Intellectual Disability – Diagnosis and Treatment*. https://doi.org/10.6000/2292-2598.2022.10.05.5

Selby, C. (2017). Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. https://doi.org/10.1145/2818314.2818315

Shute, V.J., Sun, C. & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142-158. https://doi.org/10.1016/j. edurev.2017.09.003

Vuorikari, R., Kluzer, S., & Punie, Y. (2022). DigComp 2.2: The Digital Competence Framework for Citizens - With new examples of knowledge, skills and attitudes. *JRC Publications Repository*. https://publications.jrc.ec.europa.eu/repository/handle/JRC128415 https://dx.doi.org/10.2760/115376

Wing, J. (2006). Computational thinking. Communications of the ACM, *49* (3), 33-35. https://doi.org/10.1145/1118178.1118215

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Developing *Alice*: A Scaffolding Agent for AI-Mediated Computational Thinking

Muhammad ALI[1*], Bixia CHEN[2], Gary K.W. WONG[3]
[1,2,3]The University of Hong Kong, Hong Kong SAR
akula@connect.hku.hk, s1134624@alumni.eduhk.hk, wongkwg@hku.hk

Abstract: *Recent advances in generative AI have increasingly highlighted the transformative potential of large language models (LLMs) within educational contexts. Nevertheless, the token-based generation characteristic of LLMs often results in responses that may lack depth, thereby potentially limiting their effectiveness as scaffolding tools. This paper introduces Alice, a scaffolding agent designed to provide unsolicited hints and adaptive support in computational thinking (CT) education. Alice's effectiveness was primarily evaluated through user feedback scores and benchmarked programming tasks, while further empirical research is underway to explore qualitative evidence of scaffolding effectiveness. Alice was initially optimized for both plugged and unplugged learning scenarios using a structured system prompt informed by a hierarchical framework for AI-mediated CT and the oDSP-HF approach. Subsequent fine-tuning with a LoRA-based method reduced perplexity from 9.5 to 6.6 and improved JavaScript-to-block-based code conversion accuracy from 45.8% to 69.5%. User ratings also increased from 64% to 85%. These findings tentatively indicate that careful system prompt design, combined with targeted fine-tuning, may enhance the adaptive support and learner engagement provided by LLM-based scaffolding agents in CT education.*

Keywords: computational thinking, artificial intelligence (AI), scaffolding agent, large language models, fine-tuning

1. Introduction

The rapid advancement of generative AI (GenAI) has arguably garnered significant interest in its potential to enhance personalized learning, instructional design, and assessment (Khosravi et al., 2023). Within this context, a key area of scholarly exploration concerns AI-powered scaffolding agents that dynamically adapt instructional content and provide real-time, personalized hints—often referred to as "unsolicited scaffolds" (Hijón-Neira et al., 2023). However, despite such promise, substantial challenges arise from the inherent limitations of GenAI technologies, particularly large language models (LLMs), which operate primarily through token prediction rather than explicit reasoning (Wei et al., 2022). While LLMs appear to demonstrate strong language generation capabilities, they are also prone to hallucinations and frequently lack contextual awareness. Unlike human writers, LLMs do not autonomously seek clarification on task requirements, a limitation that has led Hicks et al. (2024) to characterize them as "bullshit generators." Although prompt engineering techniques, such as "CoT prompting," have been developed to mitigate these issues (Wei et al., 2022), current evidence suggests that such strategies remain largely insufficient for providing meaningful scaffolding.

Beyond technical limitations, the developmental trajectory of LLMs further complicates their role as scaffolding agents. As these models become increasingly optimized for information retrieval rather than interactive learning (Khosravi et al., 2023), they typically provide direct answers instead of engaging in inquiry-based scaffolding. Effective scaffolding, however, arguably requires assessing prior knowledge, posing counter-questions, and fostering metacognitive engagement—pedagogical affordances that are often absent in current LLM interactions. Chen et al. (2023) emphasize the need for AI-powered scaffolding agents to align with pedagogical affordances, including reflectivity and effortful engagement, thereby necessitating both new human competencies and safeguards against unintended consequences. This context raises the first critical research question (**RQ1**): **How can AI-powered scaffolding agents be effectively developed to align with key pedagogical affordances?** One potential approach involves designing structured "system prompts" to guide LLMs in scaffolding interactions (Zhang et al., 2024). Nevertheless, research on translating prompt engineering into practical educational applications remains limited.

These challenges become even more pronounced at the intersection of AI and computational thinking (CT) education. CT has long been recognized as a crucial problem-solving skillset, with its conceptual roots in Seymour Papert's work (Lodi & Martini, 2021). While Jeannette Wing popularized the term, CT remains an evolving construct shaped by multiple frameworks, which arguably necessitates a reconsideration of Papert's foundational ideas to adapt them to emerging educational needs (Wong et al., 2020). Papert's constructionist vision emphasized hands-on engagement with computational artifacts as a means of fostering problem-solving skills, positioning CT as a mindset for leveraging computational technologies—including AI—to explore and solve problems. Despite early synergies between CT and AI, it may be observed that modern discourse has yet to fully articulate their relationship. Given AI's increasing presence in education, clarifying how it mediates CT learning while preserving human cognitive processes remains a pressing concern.

One area in which GenAI has been explored in CT education is programming-related learning activities. Researchers have integrated LLMs into block-based coding environments to support CT instruction, aligning with Papert's emphasis on active engagement with computational artifacts (Ali et al., 2024). LLMs may assist students in understanding computational concepts and generating text-based code, yet notable challenges persist. For instance, Kong et al. (2024) indicate that LLMs often struggle to accurately describe CT concepts, and students frequently misinterpret AI-generated code. Despite these limitations, LLMs arguably hold potential as cognitive scaffolds in CT education, particularly in programming contexts.

In response to these developments, scholars have called for new CT frameworks that integrate AI to support computational problem-solving (Ali et al., 2024; Wong et al., 2020). To encapsulate this evolving relationship, we propose 'AI-mediated CT' as a hierarchical framework that seeks to extend Papert's vision by positioning AI as a cognitive tool that supports—rather than supplants—human cognition. AI-mediated CT reinforces the constructionist principle that learners should remain active agents in their own learning process. Within this framework, AI functions as a cognitive scaffold, offering hints, feedback, and adaptive support to enhance computational problem-solving.

This conceptualization gives rise to the second critical research question (**RQ2**): **How can AI-mediated CT be pragmatically structured as a framework to guide system prompt design for LLMs?** Addressing this question necessitates a logically organized framework that accommodates both plugged (programming-related) and unplugged (non-programming-related) learning scenarios. The following section delineates the conceptual foundations of the research project, thereby establishing the basis for the development of *Alice*—the AI-mediated CT scaffolding agent.

2. Conceptual Groundwork

2.1. Optimizing Directional Stimulus Prompting Through Human Feedback: A Structured Approach

This subsection addresses **RQ1** in part by drawing upon contemporary technical research to propose a structured approach for designing system prompts that enables LLMs to exhibit key pedagogical affordances essential for effective scaffolding. As previously highlighted, such affordances include promoting reflectivity, fostering metacognition, and encouraging effortful engagement in learner interactions (Chen et al., 2023). In this context, system prompts refer to embedded instructions that persist in the background each time an LLM is initialized for a new inquiry, thereby defining the scope of interaction without altering the model's internal parameters (Zhang et al., 2024).

It may be argued that system prompts can be strategically designed to achieve these pedagogical affordances by instructing LLMs to avoid providing direct answers. Instead, these prompts facilitate a controlled conversational process involving hint generation, meaningful questioning, iterative refinement based on user feedback, and collaborative solution derivation. This structured approach, delineated here as *Optimizing Directional Stimulus Prompting Through Human*

Feedback (oDSP-HF), builds upon the "hint generation" concept in Directional Stimulus Prompting (DSP) proposed by Li et al. (2023) for language summarization tasks.

Unlike DSP, which relies on a secondary model, oDSP-HF operates using only a single LLM, integrating direct human interaction to refine hint generation. Upon receiving an inquiry, the LLM generates preliminary hints, which are then iteratively refined through user feedback until a clear problem-solving direction emerges. Once the user is satisfied, the LLM synthesizes the refined hints with the original inquiry to generate the final response. This iterative exchange is posited to foster a collaborative problem-solving dynamic, thereby enhancing both guidance and understanding.

Although oDSP-HF appears to hold potential for a variety of scaffolding tasks beyond AI-mediated CT, its implementation in this context introduces unique complexities. Unlike language summarization tasks, hint generation for AI-mediated CT arguably requires a more nuanced approach to system prompting. The following subsection, therefore, explicates a hierarchical framework designed to guide system prompting with oDSP-HF, thereby ensuring better alignment with the expectations of AI-mediated CT.

2.2. A Hierarchical Framework for AI-Mediated CT

This subsection addresses **RQ2** by expanding on the conceptualization of AI-mediated CT discussed in Section 1. AI-mediated CT frames AI as a scaffolding agent that provides unsolicited hints based on human feedback, augmenting computational reasoning and supporting computational problem-solving. However, as previously noted, CT—let alone *thinking* itself—is not as well-defined as strictly outcome-driven tasks such as language summarization. Thinking is highly dynamic and varies across individuals (Shin, 2019), making it inherently more complex to scaffold effectively.

A pragmatic approach to addressing this challenge is to refine the focus from 'thinking' to 'reasoning.' While thinking encompasses a broad set of cognitive processes, reasoning relies on logic or structured rules to draw inferences (Shin, 2019). By approximating human reasoning, an AI may, in principle, scaffold human thinking to a practical extent. This suggests that thinking can be conceptualized as a hierarchy of reasoning complexity, ranging from general information retrieval to contextualized problem-solving. However, when applied to CT, the absence of a structured framework becomes evident. To the best of our knowledge, no existing framework explicitly enables AI to approximate CT, even hypothetically. While multiple interpretations of CT exist, they vary in alignment and offer holistic perspectives on key computational concepts such as algorithms, decomposition, iteration, abstraction, and debugging. Yet, an epistemological gap remains in understanding how these concepts interrelate with computational reasoning and problem-solving.

To bridge this gap, *three* CT experts in computer science, AI, and education systematically reviewed existing frameworks and identified overlapping dimensions, including only constructs with 100% interrater agreement. This rigorous process underscored the need for a dynamic, hierarchical framework applicable to both plugged and unplugged learning contexts. The resulting framework comprises *four* key dimensions, namely Computational Reasoning, Computational Concepts, Computational Practices, and Computational Constructs, as illustrated in Figure 1.

At the base of this hierarchy is ***Computational Reasoning***, which is argued to be a cornerstone of CT. Computational reasoning relies on formal logic to represent computational problems, address them systematically, and derive conclusions (Paulson, 2018). Researchers assert that LLMs exhibit emergent abilities to approximate human reasoning, particularly computational reasoning, to a noticeable extent (Wei et al., 2022). Therefore, computational reasoning serves as a crucial link between AI and CT, positioning it as the fundamental dimension for integrating the hierarchical framework with the oDSP-HF approach to guiding the system prompting of LLMs for scaffolding AI-mediated CT.

The next level, ***Computational Concepts***, applies computational reasoning to general problem-solving contexts. Concepts such as algorithms, decomposition, iteration, and abstraction become relevant at this stage (Ali et al., 2024). Since different CT frameworks define these concepts in varying ways, there is no fixed set of computational concepts. Following this, the hierarchy progresses to ***Computational Practices***, which involve applying computational concepts to specific problem-solving scenarios. For example, decomposition—the process of breaking down a problem into smaller parts—is a computational concept. However, when applied in a structured problem-solving task, decomposition becomes a computational practice (Ali et al., 2024). As a computational practice in the development of a pathfinding algorithm, 'decomposition' involves breaking the problem into subproblems such as graph representation, node traversal, and cost evaluation. Each subproblem is addressed individually before being integrated into a complete solution.

The final level, ***Computational Constructs***, represents the implementation of computational practices within programming environments. Iteration, for instance, functions as both a computational concept and a computational practice, involving the repeated execution of a process until a condition is met (Ali et al., 2024). However, when implemented in programming, iteration is operationalized through constructs such as FOR loops, WHILE loops, or recursive function calls. In search algorithms, these constructs provide the syntactic and structural mechanisms necessary to execute iteration in code. Thus, 'iteration' transitions from a computational practice to a formalized computational construct within a programming language.

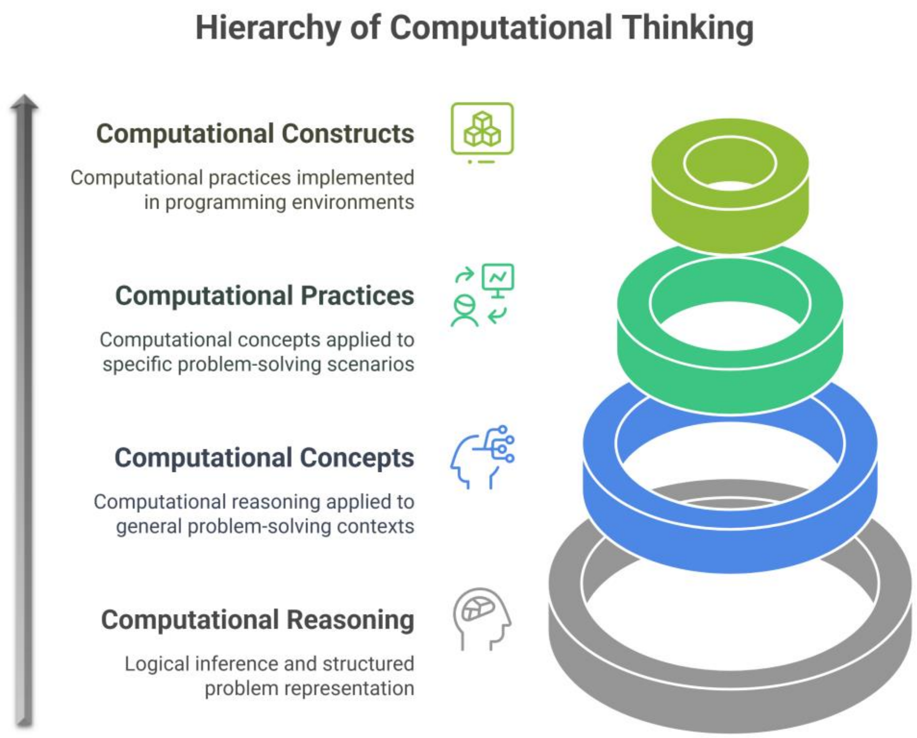


Figure 1. The hierarchical framework for AI-Mediated CT.

Although not intended as a definitive model, this pragmatic framework (see Figure 1) serves to structure AI-mediated CT for system prompt design and scaffolding. The following section outlines *Alice*'s technical implementation using the oDSP-HF approach, which is informed by the hierarchical framework, as well as the subsequent fine-tuning and performance evaluation.

3. Technical Implementation

Alice, the AI agent, is built upon Llama 3.1 (70B), an open-source LLM developed by Meta. This model was selected for its adaptability, its support for fine-tuning, and its competitive long-context reasoning capabilities, all of which are considered essential for structured system prompting and are understood to align closely with proprietary models such as GPT-4.

For the initial implementation (**RQ1**), the system prompt was designed following the oDSP-HF approach, as guided by the hierarchical framework for AI-mediated CT. The prompt was crafted to specify *Alice*'s persona, incorporate DSP instructions, and include few-shot CoT exemplars to reinforce computational reasoning. To ensure contextually relevant hints for both plugged and unplugged learning scenarios, the system prompt was layered according to the four hierarchical dimensions of AI-mediated CT. *Alice* was deployed using Poe's server deployment feature (https://creator.poe.com/docs/), and hosted on a local server. This approach was intended to provide greater control over deployment, data management, and iterative refinements.

Alice's response accuracy, logical coherence, and adherence to the system prompt were systematically assessed following deployment. Over a period of three months, 64 K–12 trainee teachers enrolled in a CT education course at a prominent university in Hong Kong interacted with *Alice* via Poe. The course encompassed both plugged learning scenarios (e.g., Micro-Bit coding) and unplugged learning scenarios (e.g., LEGO patterns, mind mapping). In accordance with Poe's data privacy policy, only user interactions (prompt-response pairs) were recorded, with all personal identifiers excluded. This process yielded a total of 10,346 interactions, and user ratings were collected through Poe's "thumbs-up/thumbs-down" feedback system.

For the fine-tuning phase, interactions were categorized into four groups: positively rated, negatively rated at the end of conversations, neutrally rated at the end of conversations, and remaining interactions. It is notable that over 84% of negative interactions involved Micro-Bit JavaScript-to-block-based code conversion issues, which appears consistent with prior research indicating that LLMs are not specifically fine-tuned for such tasks (Kong et al., 2024). A dataset of 3,661 prompt-response pairs was curated and manually labeled for fine-tuning, which was conducted using Low-Rank Adaptation (LoRA) via the Unsloth library (https://docs.unsloth.ai/). LoRA, a parameter-efficient fine-tuning method (Hu et al., 2021), is designed to reduce computational overhead while maintaining model performance. The fine-tuning process employed a LoRA rank of 16, a LoRA alpha of 16, and a LoRA dropout of 0.1, targeting key projection modules.

Training was performed on dual RTX 3090 Ti GPUs using the AdamW optimizer, with a batch size of 2, gradient accumulation steps of 10, a learning rate of $1.5e^{-4}$, and 3 epochs, completing in 28 hours. Following fine-tuning, a validation set of 550 samples, including 25% Micro-Bit-specific queries, was used to evaluate performance. Results from *Alice*'s fine-tuning are summarized in Table 1.

Table 1. *Alice*'s performance pre- and post-fine-tuning.

Metric	Pre-Fine-Tuning	Post-Fine-Tuning
Perplexity (validation set)	9.5	6.6
Code conversion accuracy (all, %)	45.8 (±3.1)	69.5 (±2.5)
Code conversion accuracy (unseen, %)	37.2 (±3.4)	64.8 (±2.7)
User thumbs-up rate (%)	64	85

As indicated in Table 1, fine-tuning led to substantial improvements in *Alice*'s performance. The reduction in perplexity suggests enhanced fluency and coherence, while the marked increase in JavaScript-to-block-based code conversion accuracy and user thumbs-up rates arguably indicates more effective scaffolding. These improvements were particularly pronounced in previously unseen problems, highlighting *Alice*'s strengthened generalization capabilities. These results indirectly support the effectiveness of the oDSP-HF approach and the hierarchical framework in guiding AI-mediated CT scaffolding.

4. Conclusion

In evaluating *Alice*'s AI-mediated CT scaffolding performance, it should be emphasized that no direct quantitative metrics were available; rather, user feedback trends served as indirect performance indicators over time. To gain a deeper understanding of these observed improvements, an interpretive inquiry is currently underway to explore *Alice*'s integration within CT teacher education settings. While preliminary findings underscore the potential of AI-powered scaffolding in CT education, they also draw attention to ongoing challenges related to agent adaptability and the contextual specificity required for effective implementation. Nevertheless, our ongoing research continues to focus on refining the oDSP-HF approach to system prompting, with the broader aim of advancing the development of more sophisticated scaffolding agents capable of supporting AI-mediated CT and related educational applications.

References

Ali, M., Wong, G. K.-W., & Ma, M. (2024). K–12 Pre-service Teachers' Perspectives on AI Models and Computational Thinking: The Insights from an Interpretative Research Inquiry. *Proceedings of the 8th APSCE International Conference on Computational Thinking and STEM Education (CTE-STEM 2024)*, *8*(1), 66-71. https://doi.org/10.5281/zenodo.11559685

Chen, B., Zhu, X., & Díaz del Castillo H, F. (2023). Integrating generative AI in knowledge building. *Computers & Education: Artificial Intelligence*, *5*, 100184. https://doi.org/10.1016/j.caeai.2023.100184

Hicks, M. T., Humphries, J., & Slater, J. (2024). ChatGPT is bullshit. *Ethics and Information Technology*, *26*(2), 38. https://doi.org/10.1007/s10676-024-09775-5

Hijón-Neira, R., Connolly, C., Pizarro, C., & Pérez-Marín, D. (2023). Prototype of a Recommendation Model with Artificial Intelligence for Computational Thinking Improvement of Secondary Education Students. *Computers (Basel)*, *12*(6), 113. https://doi.org/10.3390/computers12060113

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *arXiv [cs.CL]*. https://doi.org/10.48550/arXiv.2106.09685

Khosravi, H., Viberg, O., Kovanovic, V., & Ferguson, R. (2023). Generative AI and Learning Analytics. *Journal of Learning Analytics*, *10*(3), 1-6. https://doi.org/10.18608/jla.2023.8333

Kong, S.-C., Sit, E. C. Y., Yang, N. Y., & Yeung, W. K. (2024). One Step Forward towards the Use of Human Language to Instruct ComputerstoWork: A Reflection on an Example of Applying Prompts in Text-based GenerativeAI for Programming. In *Proceedings of the 8th APSCE International Conference on Computational Thinking and STEM Education (CTE-STEM 2024)* (pp. 151-153), Beijing, China.

Li, Z., Peng, B., He, P., Galley, M., Gao, J., & Yan, X. (2023). Guiding Large Language Models via Directional Stimulus Prompting. *arXiv [cs.CL]*. https://doi.org/10.48550/arXiv.2302.11520

Lodi, M., & Martini, S. (2021). Computational Thinking, Between Papert and Wing. *Science & Education*, *30*(4), 883-908. https://doi.org/10.1007/s11191-021-00202-5

Paulson, L. C. (2018). Computational logic: its origins and applications. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *474*(2210), 20170872-20170872. https://doi.org/10.1098/rspa.2017.0872

Shin, H. S. (2019). Reasoning processes in clinical reasoning: from the perspective of cognitive psychology. *Korean Journal of Medical Education*, *31*(4), 299-308. https://doi.org/10.3946/kjme.2019.140

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS 2022)*, *36*(1), 24824-24837. https://dl.acm.org/doi/10.5555/3600270.3602070

Wong, G. K.-W., Ma, X., Dillenbourg, P., & Huan, J. (2020). Broadening artificial intelligence education in K-12: Where to start? *ACM Inroads*, *11*(1), 20-29. https://doi.org/10.1145/3381884

Zhang, L., Ergen, T., Logeswaran, L., Lee, M., & Jurgens, D. (2024). SPRIG: Improving Large Language Model Performance by System Prompt Optimization. *arXiv [cs.CL]*. https://doi.org/10.48550/arXiv.2410.14826

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Envisioning Computational Thinking Education: An Idealized Design Approach from Teachers' Perspective

Ali HAMIDI*
Linnaeus University, Sweden
ali.hamidi@lnu.se

Abstract: *This study explores the conceptualization of an idealized design for computational thinking (CT) education from teachers' perspectives, guided by the principles of interactive planning. While CT education has received significant attention, the possibility of envisioning its desired state remains underexplored. Through workshops and discussions with teachers, this study follows an idealized design approach. It involves identifying challenges in the current state of CT education, examining worst-case scenarios if these challenges persist, and exploring the possibility of envisioning a desired current state for CT education. The findings reveal not only a lack of understanding of CT but also other challenges such as inequitable access to resources and insufficient collaboration among educators. Additionally, although potential worst-case scenarios and ideal CT education are complex to conceptualize, key themes emerged. Worst-case scenarios included a widening achievement gap, diminished problem-solving skills, and reduced motivation among advanced learners. Meanwhile, the envisioned ideal CT education encompassed themes such as skills and competencies, equity and access, and the transformative concept of School 2.0. These themes were subsequently positioned within cognitive, situated, and critical framings of CT to better contextualize the findings and connect them with existing research. This framing helped illustrate the coherence of these themes while broadening the scope of CT education research.*

Keywords: Computational thinking, systems thinking, idealized design, interactive planning

1. Introduction

Computational thinking (CT) education has been extensively researched from diverse perspectives, encompassing curriculum content, educational contexts, pedagogical approaches, and assessment methodologies. Each perspective provides insights into different aspects of CT education, especially as interest in the topic has grown over the past two decades. However, the idealized perspective of CT education remains underexplored. Previous studies have examined the development of CT education by focusing on cognitive and societal dimensions (Kafai et al., 2020; Mills et al., 2021) and aim to advance CT as a foundation for contemporary education and a data-driven society (Dolgopolovas & Dagiene, 2021; Tedre et al., 2021). However, these approaches are predominantly grounded in conventional planning frameworks such as reactivism and preactivism, which correspond to past and future-oriented planning approaches (Ackoff, 2001). This study employs an alternative approach, known as idealized design, rooted in interactive planning. It aims to explore teachers' visions of an ideal current state of CT education through a structured systemic approach. By reflecting on existing challenges and potential worst-case scenarios, this study seeks to answer the following question: How and to what extent can an idealized CT education be conceptualized from teachers' perspectives?

The paper is structured as follows: first, the background outlines key literature and concepts. Next, the theoretical approach, settings, and context are presented, followed by the empirical findings, discussion, and conclusion.

2. Background

CT involves mental skills and practices for designing computations and interpreting the world as a system of information processes (Denning & Tedre, 2019). It has evolved conceptually and practically, with various frameworks to define its principles and implications. While many define CT through components such as abstraction, algorithms, patterns, and decomposition (Brennan & Resnick, 2012; Grover & Pea, 2013), others take alternative approaches. For instance, Weintrop et al. (2016) frame CT as a taxonomy that highlights practices such as modeling and simulation, data-related practices, computational problem-solving, and systems thinking.

From a broader perspective, Kafai et al. (2020) proposed a three-layer framing of CT, offering new development directions with a humanistic approach. The core layer, cognitive CT, focuses on individual learning and skill acquisition for problem-solving. The middle layer, situated CT, connects CT to personal interests, social interactions, and identity formation, emphasizing collaborative learning. The outer layer, critical CT, addresses the societal and ethical implications of computing by analyzing power structures and promoting justice and inclusivity in problem-solving.

To ensure CT education is both meaningful and sustainable, it should be viewed through the lens of systems thinking. Idealized education systems offer a useful framework for this purpose. Research on idealized design in education is limited, as the concept primarily originates from management and product development (Ackoff & Rovin, 2003). However, Russell Ackoff, who introduced this approach, emphasized its relevance for improving educational contexts, encouraging a systemic rethinking of learning environments (Ackoff & Greenberg, 2008). It was argued that educational shortcomings stem from systemic issues rather than isolated inefficiencies. The idealized design has also been applied to education as an alternative to traditional strategic planning, representing its potential to enhance stakeholder engagement and improve systemic decision-making (Pickering, 2006).

Envisioning an ideal educational system that fosters CT is challenging due to the ambiguity surrounding CT (Shute et al., 2017) and the complexity of its integration into educational contexts. Nevertheless, recent CT frameworks (Kafai et al., 2020; Mills et al., 2021) implicitly address the characteristics of an ideal CT education, particularly emphasizing societal, ethical, and inclusivity aspects. The idealized design approach, which involves envisioning an "ideal current state" aligns with these frameworks by conceptualizing CT as a multimodal language, an advanced educational skill, a mental tool for problem-solving, and a pathway toward artificial intelligence (AI) thinking (Dolgopolovas & Dagiene, 2021).

3. Theoretical Approach

The primary theoretical lens guiding this study is idealized design, an approach rooted in systems thinking. Systems thinking tools, such as rich pictures and causal loop diagrams, have also been used in workshops to help participants visualize the broader landscape of CT education and analyze interconnections and influences among its components. Idealized design refers to the process of designing a system that its designers would choose to implement if they had the autonomy to select any system they desired (Ackoff, 2001). It envisions the "best possible" replacement while considering constraints such as technical feasibility, operational practicality, and adaptability to change (Ackoff & Rovin, 2003). Idealized design is the central concept in interactive planning, which consists of two main processes: idealization and realization, encompassing six stages ranging from formulating the mess to designing controls. This research focuses only on the idealization phase, specifically formulating the mess and ends planning, as illustrated in Figure 1.

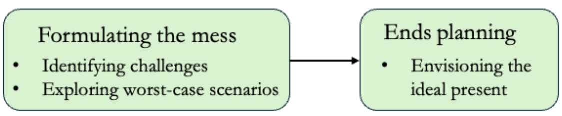


Figure 1. Simplified idealized design approach applied in the research

The idealization phase involves identifying challenges, exploring worst-case scenarios, and envisioning the ideal present. The first two activities focus on the process of formulating the mess, incorporating the use of rich pictures and causal loop diagrams. The third activity, in contrast, is dedicated to ends planning.

4. Settings and Context

The data for this study were collected in two settings in Sweden, involving teachers from grades 7 to 9. The first setting consisted of a two-session workshop held at the university, while the second took place at a school and included two workshops and two discussion sessions. At the university, two workshops (each lasting two hours) involved five teachers and one pedagogue from six different schools. At the school, two workshops and two discussion sessions (each lasting 45 minutes) included 15 teachers. The sessions began with a video introduction that provided an abstract overview of CT and idealized design. Educational robotics and drones were also showcased to motivate teachers and familiarize them with digital technologies related to CT, offering inspiration and supporting their involvement in idealization practices. Drawing rich pictures and causal loop diagrams were supplementary methods. These helped teachers map the current state of CT education, its integrated components, and potential scenarios and outcomes if existing strategies and approaches continue.

The data collection method involved recording the discussions in each group and taking notes by researchers who participated in the sessions. The recorded files were later transcribed and translated into English to facilitate analysis and derive results. The transcribed data were then thematically analyzed and structured according to the three steps outlined in the theoretical approach (see Figure 1), aligning with the study's objectives. Consent for recording the discussions and using the data for research purposes was obtained from all participants in advance.

5. Findings

5.1. Teachers' Perceptions of CT and Challenges

Teachers had a diverse understanding of CT, with some associating it with programming and AI, while others emphasized structured problem-solving, as one teacher explained, "it's about solving problems in a structured way". Many implicitly linked CT to critical thinking, highlighting pattern recognition, abstraction, and collaboration as key aspects. Some expanded the definition to include general computer skills, creativity, and workflow development, noting that students tend to rely on pre-existing structures rather than developing their own.

Key challenges from teachers' perspectives included a lack of institutional support, collaboration, and resources, forcing teachers to integrate CT independently. One teacher noted, "each teacher decides for themselves how to deal with these issues", while another described the isolation, saying, "it's like everyone is on their own island". Limited training, unclear guidelines, and resistance to change further hindered implementation, with one teacher stating, "there's a lot of hesitation to change the way things are done, even if it's for the better".

5.2. Picturing Potential Worst-case Scenarios

The workshop discussions provided several potential worst-case scenarios if the current trends in CT integration persist in the educational context, particularly in Sweden. Teachers expressed concerns that declining critical thinking skills among students could weaken their problem-solving abilities, ultimately leaving graduates unprepared for the demands of society. Another major issue was the growing divide between high and low-performing students, with only those performing well benefiting in the long run. As one teacher remarked, "this divide will be increased and increased, and finally, it would be only the high-performance students who probably in the future get the job positions".

Additionally, they worried that talented students might lose motivation due to a lack of challenges, as one teacher observed, "we have a group of students who are very smart, but they don't get the extra challenges they need. This risks killing their interest". Lastly, policy decisions, such as the push for "screen-free schools", were seen as conflicting with the increasing digitalization of education. Teachers also warned of potential economic consequences, predicting that companies might have to hire skilled workers from abroad, which could weaken the country's global competitiveness in technology and innovation.

5.3. Envisioning Ideal Current

Teachers found it difficult to clearly define an ideal CT education, as their vision extended beyond CT to encompass broader skills and competencies. Rather than focusing solely on CT education, they emphasized its wider impact on individual and societal development. Problem-solving emerged as the most frequently mentioned skill, with teachers envisioning students applying it across different contexts in an ideal system. They also highlighted the importance of identifying and nurturing individual talents and interests, ensuring that students receive relevant learning resources to develop their strengths. Additionally, collaboration and teamwork were seen as crucial, allowing students to leverage each other's strengths and learn collectively.

Equitable access to resources was another key feature of an ideal CT education system, ensuring that all students, regardless of school size or location, have equal opportunities to learn CT skills. Teachers envisioned a system that encourages creativity and innovation, fostering out-of-the-box thinking and the development of new solutions. Some teachers envisioned a "School 2.0" model based on four dimensions of society, resources, knowledge, and competency. It emphasizes stakeholder collaboration, accessible learning resources, transparent assessments, and skill development aligned with emerging trends through partnerships with industries, universities, and schools.

6. Discussions

6.1. Where Today's CT Education Fails

Understanding of CT is likely the primary factor influencing its integration into education. A lack of understanding, especially among teachers, poses an initial challenge, leading to uncertainty about how to integrate it effectively (Ling et al., 2017). When CT integration is viewed as a system of interconnected components (Hamidi, 2025), misunderstanding or lack of understanding becomes a key factor influencing the rest. Although some teachers recognized CT, most lacked a clear understanding, which led discussions to focus more on broader competencies such as problem-solving, collaboration, and digital literacy. While these align with 21st-century skills, limited attention to CT-specific competencies weakens its impact. This gap in understanding hinders effective implementation and reduces CT's potential to develop critical and analytical thinking.

In addition to misunderstanding, teachers identified several worst-case scenarios that could arise if CT education remains poorly integrated. These included a growing gap between high- and low-performing students, a decline in core cognitive skills, increased individualism, student disengagement, and adverse economic effects. Ackoff (2001) describes such scenarios as reference projections, offering foresight into how a system could destroy itself. To better understand these challenges within the context of CT education, they are analyzed through the cognitive, situated, and critical framings proposed by Kafai et al. (2020). This approach offers a systemic perspective that extends beyond individual cognition. As illustrated in Figure 2, these issues are placed in the lower half of the original model to represent potential failures that could undermine the goals of CT education.

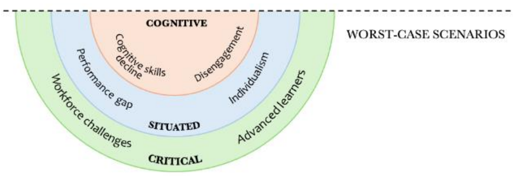


Figure 2. Cognitive, situated, and critical contexts for worst-case scenarios in CT education

The cognitive framing encompasses problem-solving deficiencies and disengagement, which hinder the development of analytical skills (Grover & Pea, 2013). The situated framing relates to educational inequities, as the achievement gap challenges inclusivity in CT education (Liu, 2023). Additionally, individualism and lack of collaboration highlight the absence of CT's social dimensions. Finally, the critical framing reflects the broader societal consequences, including student disengagement from meaningful applications of CT and economic consequences due to a lack of skilled graduates (Kafai et al., 2020).

6.2. Crafting the Current Ideal

Envisioning an ideal CT education was challenging for teachers, as their responses covered both relevant and broader educational aspects. However, their ideas aligned with three key themes: skills and competencies, equity and access, and the transformative concept of School 2.0.

The first key theme focuses on skills and competencies, where teachers envisioned students applying problem-solving across diverse contexts. This involves the discovery and development of individual talents and interests, enabling students to pursue personalized learning paths that align with their interests. Collaboration and teamwork were also emphasized, as leveraging collective strengths fosters deeper learning. Additionally, creativity and innovation were seen as essential in enabling students to develop new solutions and products. The second theme, equity and access, emphasizes the importance of providing equal opportunities to teach and learn CT, regardless of school context. This aligns with Ackoff and Greenberg's (2008) view of an ideal learning environment as one where both teachers and students learn from one another, supporting more inclusive and collaborative educational systems. The third theme introduces School 2.0, a transformative model integrating societal, resource, knowledge, and competency aspects into CT education.

Similar to worst-case scenarios, the envisioned ideal CT education can be analyzed through the cognitive, situated, and critical framings of CT (Kafai et al., 2020). Figure 3 illustrates this by representing the core aspects of the ideal envisioned CT education, which is positioned in the upper section of Kafai et al.'s (2020) model. This contrasts with the worst-case scenarios placed in the lower section, as previously shown in Figure 2.

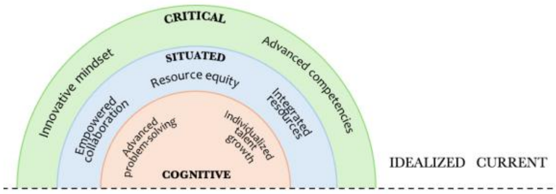


Figure 3. Cognitive, situated, and critical contexts for idealized CT education

Problem-solving and individualized learning align with cognitive CT, ensuring students develop analytical and reasoning skills. Collaboration and equitable access fit within situated CT, emphasizing inclusive learning environments. Finally, creativity, innovation, and adaptability correspond to critical CT, focusing on real-world problem-solving and workforce readiness. This framework ensures that teachers' envisioned CT education aligns with broader educational objectives while remaining practical and adaptable.

7. Conclusion and Future Work

This study applied an idealized design approach to examine how and to what extent an idealized CT education can be conceptualized from teachers' perspectives. Key challenges identified such as a lack of CT understanding, and inequitable resource distribution, leading to potential worst-case scenarios, for example, widening educational disparities and declining problem-solving skills. Despite the complexity of envisioning an ideal CT education, three core themes emerged: skills and competencies, equity and access, and the transformative model of School 2.0. These themes were framed within cognitive, situated, and critical perspectives, aligning with and expanding existing CT research. Given the study's limited sample size and the absence of participants with prior CT teaching experience, future research should engage a broader and more diverse group. Applying a more comprehensive idealized design approach that includes the realization phase could further refine ideal CT education and lead to more effective implementation in practice through specific classroom applications and actionable strategies.

References

Ackoff, R., & Rovin, S. (2003). *Redesigning Society*. Stanford, CA: Stanford University Press.

Ackoff, R. (2001). *A Brief Guide to Interactive Planning and Idealized Design*. https://www.ida.liu.se/~steho87/und/htdd01/AckoffGuidetoIdealizedRedesign.pdf

Ackoff, R., & Greenberg, D. (2008). *Turning Learning Right Side up: Putting Education Back on Track*. Pearson Prentice Hall. https://www.nigerianseminarsandtrainings.com/articles-pdf/47.02.TurningLearning.pdf

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).

Denning, P. J., & Tedre, M. (2019). *Computational Thinking*. Mit Press.

Dolgopolovas, V., & Dagiene, V. (2021). On the future of computational thinking education: Moving beyond the digital agenda, a discourse analysis perspective. *Sustainability, 13*(24), 13848. https://doi.org/10.3390/su132413848

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher, 42*(1), 38-43. https://doi.org/10.3102/0013189X12463051

Hamidi, A. (2025). Advancing computational thinking education: Insights from systems thinking applications. *Human Systems Management, 44*(1), 157-172. https://doi.org/10.3233/HSM-240024

Kafai, Y., Proctor, C., & Lui, D. (2020). From theory bias to theory dialogue: embracing cognitive, situated, and critical framings of computational thinking in K-12 CS education. *Acm Inroads, 11*(1), 44-53. https://doi.org/10.1145/3381887

Ling, U. L., Saibin, T. C., Labadin, J., & Aziz, N. A. (2017). Preliminary investigation: Teachers' perception on computational thinking concepts. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9*(2-9), 23-29. e-ISSN: 2289-8131

Liu, Z. L. (2023). Integrating computational thinking into K-12 education: Bridging the gap between theories and practices. *STEM Education Review.* http://dx.doi.org/10.54844/stemer.2023.0467

Mills, K., Coenraad, M., Ruiz, P., Burke, Q., & Weisgrau, J. (2021). *Computational Thinking for an Inclusive World: A Resource for Educators to Learn and Lead*. Digital Promise. https://doi.org/10.51388/20.500.12265/138

Pickering, T. D. (2006). *Redesigning public school systems: A qualitative study into the potential for utilizing idealized design to create the "next generation" of American public schools*. Drake University.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational research review, 22*, 142-158. https://doi.org/10.1016/j.edurev.2017.09.003

Tedre, M., Denning, P., & Toivonen, T. (2021, November). CT 2.0. In *Proceedings of the 21st Koli Calling International Conference on Computing Education Research* (pp. 1-8). https://doi.org/10.1145/3488042.3488053

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *J Sci Educ Technol, 25*, 127-147. https://doi.org/10.1007/s10956-015-9581-5

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Evaluation of Learning Effectiveness by Integrating Self-Directed Learning and Blended Learning: A Case Study on Bus Topology Curriculum Unit

Mei-Ling LIU[1], Shih-Hua HUANG[2], Ting-Chia HSU[3*]
[1]Kaohsiung Municipal Kaohsiung Commercial High School, Taiwan
[2,3] National Taiwan Normal University, Taiwan
ckhsu@ntnu.edu.tw

Abstract: *This study was based on Taiwan's 108 Curriculum Guidelines for Technical Senior High Schools, developing knowledge nodes and instructional videos on the Taiwan Adaptive Learning Platform. A technology-assisted self-regulated learning lesson plan was designed and implemented to integrate both online and offline learning, forming a blended learning instructional model. The learning process included self-regulated learning planning, independent study using the Taiwan Adaptive Learning Platform and WSQ worksheets, intra-group collaborative learning, inter-group peer learning, teacher-guided instruction, and final reflective worksheet writing. The bus topology unit was selected as a case study for implementation.The results showed that students' post-test scores were significantly higher than their pre-test scores, proving the effectiveness of this learning model. Additionally, students who completed the WSQ worksheets demonstrated better learning effectiveness than those who did not, suggesting that the worksheets helped enhance learning effectiveness.*

Keywords: Self-Regulated Learning, Blended Learning, Instructional Videos, Taiwan Adaptive Learning Platform, WSQ Worksheet

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

結合自主學習與混成式學習之學習成效分析：以匯流排拓樸課程單元為例

劉美伶 [1]，黃詩樺 [2]，許庭嘉 [3*]
[1] 高雄市立高雄高級商業職業學校資料處理科，高雄，臺灣
[2,3] 臺灣師範大學科技應用與人力資源發展學系，臺北，臺灣
ckhsu@ntnu.edu.tw

【摘要】 本研究搭配臺灣 108 技術高中課程綱要發展數位學習平台因材網上面之知識節點與教學影片，並且進行單元教學實證，設計科技輔助自主學習教案，整合線上和線下發展混成式學習教學設計，流程包括自主學習規劃、透過撰寫 WSQ 學習單結合數位學習平台-因材網進行自學、組內共學、組間互學、教師導學等階段並在最後撰寫反思學習單，本研究以匯流排拓樸單元為例。研究結果顯示，學生後測成績顯著高於前測，證明本學習模式能有效促進學習，完成 WSQ 學習單的學生學習成效明顯優於未完成者，顯示學習單有助於深化學習效果。

【關鍵字】 自主學習；混成式學習；教學影片；因材網；WSQ 學習單

1. 簡介

隨著科技不斷發展以及過去疫情影響，數位學習平台已經成為現代教育的重要工具，其提升了學生的學習體驗，突破了傳統教學在時間與空間上的限制，使學生能夠根據自身需求自主安排學習進度，提供更靈活、個性化與互動化的學習環境(Liu, 2024)，透過多媒體教學影片與互動式學習資源，數位學習平台不僅能加強學生對抽象概念的理解，還能提供即時回饋，促進自主學習能力的培養，特別是在技職教育領域，數位學習平台能將複雜的專業知識以視覺化方式呈現，使學生更容易掌握相關知識，Nácher 等人(2021)的研究指出，透過數位學習平台進行學習對於提升學生的學習成績有正面影響，也強調了混成式的教學對於提升學習成績的重要性。

儘管數位學習平台具備諸多優勢，但許多學生仍未能充分利用這些資源，導致學習成效不如預期 (Rakic et al., 2020)，其原因主要包括學習動機不足、自主學習能力有限、平台使用體驗不佳以及教學方法與平台功能未充分整合等因素(Rakic et al., 2020; Xie & Wei, 2020)，例如:Zhao 等人(2020)的研究指出，在疫情期間雖然學生使用數位學習平台的頻率大幅提升，但由於缺乏明確的學習目標、有效的學習策略與持續的自我管理，其學習行為並未隨之改善，Ordones and Ferneda(2022)也強調，數位學習平台雖提供多元化資源，但若未能輔以清晰的學習規劃與適切的引導，僅依賴平台本身並不足以保證學生達成理想的學習成果，因此，將數位學習與自主學習結合，不僅能幫助學生提升自我管理能力，還能增強學習動機，從而有效克服挑戰。

自主學習(Self-regulated Learning, SRL)已被廣泛應用於教育領域，並且被證實能夠有效提升學生的學習動機以及自我管理能力，在數位學習的環境中極為重要(Rakic et al., 2020; Liu, 2024)，過去研究指出，具備自主學習能力的學生更能積極參與學習並取得優異成績(Jansen et al., 2019)。在本研究中學生會撰寫 WSQ 學習單，學習單包括自主學習規劃、學習反思，自主學習規劃單明確設定學習目標選擇適合的學習策略並且安排學習進度以及時間，學習反思單檢視學習策略的成效，在發現問題時及時調整，這種結合自主學習的方式不僅能提升學生的學習動機，還能培養其自我管理能力，使學習過程更加高效且具有持續性。

除了自主學習，混成式學習(Blended Learning)也已成為現代教育的重要趨勢，混成式學習將線上與線下教學相結合，保留了傳統教室互動合作的優勢，又充分利用數位教學平台提供的靈活學習方式，實現了更高效率的教學(Ordones & Ferneda, 2022)，這種學習模式不僅能提高學生的學習參與度，還能夠促進課堂討論與同儕合作，為了進一步提升混成式學習的成效，本研究採用 WSQ 學習單，幫助學生在自主學習與課堂互動之間建立更緊密的聯繫，WSQ 學習單包含以下三個部分：Watch(觀看)：學生觀看指定的教學影片，初步理解課程內容、Summarize(總結)：觀看影片後，學生撰寫摘要，整理關鍵概念與學習重點、Question(提問)：學生根據影片內容提出至少一個問題，促進批判性思考同時為課堂討論提供有深度的議題與思考方向。通過線上與線下相結合，學生在自主學習的基礎上，在課堂中與教師及同儕進行問題討論互動，加深對於課程內容的理解及應用，WSQ 學習單作為自主學習歷程工具，不僅強化學生對教學影片內容的理解與反思，也將線上自主學習成果延伸至課堂討論，實現線上學習與實體學習的有效結合，是混成式學習中促進學習深化的重要設計之一。

綜上所述，本研究結合自主學習與混成式學習探討學生的學習成效，並以「匯流排拓樸課程單元」作為研究主題，待答問題如下：

(1)透過自主學習與混成式學習利用數位學習平台-因材網進行學習，學生的學習成效是否有顯著進步?

(2)在課程中有完成 WSQ 學習單的學生，學習成效是否有顯著高於沒有完成 WSQ 學習單的學生?

2. 文獻探討

2.1. 混成式學習(Blended Learning)

混成式學習（Blended Learning, BL）是一種將線上與實體面授教學元素相結合的教育模式，旨在融合兩者優勢，讓學生在享有靈活學習的同時，獲得即時互動與指導(Graham & Halverson, 2022)，Dziuban 等人(2018) 的研究指出，混成式學習不僅能提升學習成效，還能增強學習彈性並降低教育成本，因此被譽為教育界的「新常態」。此外，混成式學習具有學習時間與地點的靈活性，能夠滿足不同學習風格，並使學習資源更易於取得，從而提升學習效率並培養學生的自主學習能力(Oskah Dakhi et al., 2022)。

隨著教育理念的轉變，混成式學習在促進學生自主學習方面的重要性日益凸顯，Jumaini et al.(2021)的研究指出，透過多元化的學習資源與彈性的學習方式，混成式學習能夠激發學生在課堂之外自主探索知識，進而加深對概念的理解，除此之外，混成式學習還有助於培養學生的溝通、解決問題以及推理等高階認知能力(Cao, 2023)， Li & Wang (2022) 的研究進一步證實，混成式學習能顯著提升 K-12 學生在認知領域的整體學習表現，特別是在小組互動與合作學習的過程中，不僅促進了概念理解與學習參與，還增強了學生的社交技能與團隊合作能力。綜上所述，混成式學習已成為現代教育的重要模式，不僅能促進學生的自主學習，還能培養學生在合作以及溝通方面的核心素養，為不同學科與學習風格的學生提供更靈活且高效的學習途徑。

2.2. 自主學習(Self-Regulated Learning)

自主學習（Self-Regulated Learning, SRL）指的是學生透過自我調整思考、行為和情緒，主動規劃並持續努力以達成學習目標（Schunk & Zimmerman, 1994），在學習過程中，學生需設定學習目標、選擇並調整適當的學習策略、監控學習進度，並透過反思來評估學習成果，並確保達到目標的策略是否有效，在過去的許多研究均指出，自主學習能力是預測學業表現的重要因素，具備較強自主學習能力的學生通常能夠提升學習參與度，進而獲得更佳的學業

成績(Jansen et al., 2019)，Vosniadou(2020)的研究中也發現，若從高中階段開始培養自主學習能力，學生在進入大學後能更快適應其學習模式。

隨著新冠疫情促使數位學習迅速發展，在教育領域當中自主學習的重要性愈加受到重視，由於線上學習缺乏即時指導與社交互動，學習者需具備更高的自我管理能力以克服這些挑戰(Broadbent & Poon, 2015)，Bernacki 等人(2020)的研究亦指出透過線上學習環境有潛力培養學生的自主學習能力。然而，除了純線上學習外，混成式學習 (Blended Learning) 逐漸成為現代教育的重要趨勢，其結合了傳統面對面教學與數位學習資源，讓學生擁有更多元的學習體驗，Tuilan(2023) 的研究證實自主學習在混成式學習環境中對提升學業成績具有關鍵作用。

3. *研究方法*

3.1. 教學設計

本教學透過四學理論進行教學設計(如圖 1)，首先，會由學生自學開始，學生依據自主學習規劃單自訂學習目標，並且透過觀看數位平台-因材網自學匯流排拓撲的運作原理、特性與其優缺點等課程接續完成 WSQ 學習單，完成後進行組內共學階段，學生透過分組討論影片與講解內容，將關鍵點寫在白紙上，再到組間互學階段，各組選一位代表向其他各組分享小組討論總結，透過教師補充並強調重點，最後回到教師導學，確認學生是否有明確了解課程內容以及進行課程總結並引導學生完成自主學習反思單反思學習狀況。

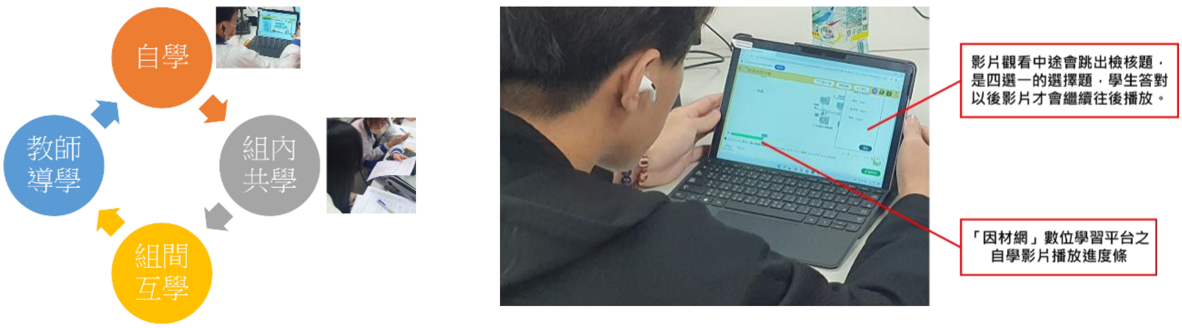


圖 1 四學教學設計(Ho, 2022)、因材網自學介面

3.2. 研究對象

本次參加因材網混成式學習的學生年齡為 17~18 歲，12 年級技術高中學生，總共參加人數為 34 人，男生 14 名、女生 20 名，都有進行因材網混成式學習，經歷四學教學設計，但是其中有 25 人有確實完成課程設計的學習單，有 9 人沒有完全完成。

3.3. 實驗流程

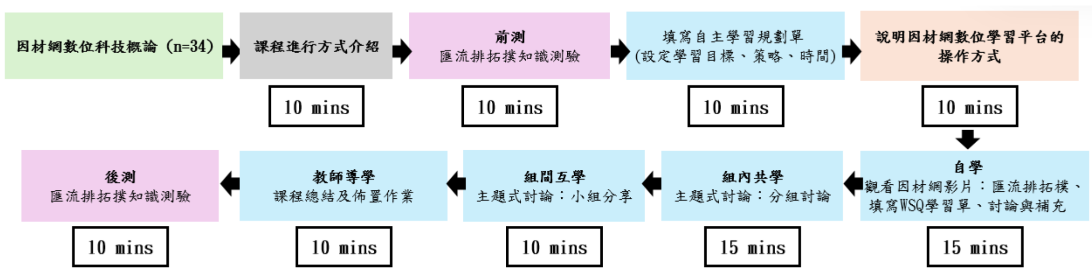


圖 2 實驗流程圖

4. 研究結果

4.1. 整體學習成效分析

學習成效前後測具有顯著的差異(p<.001)。後測成績(M =69.71；SD =24.31)顯著高於前測成績(M =42.35；SD =18.43)。結果顯示學生透過本研究自主學習與混合式學習方式利用數位學習平台-因材網進行學習，學生的匯流排拓樸課程概念有顯著提升，表示學生有學習到匯流排拓樸相關知識。統計結果如下表 1。

表 1 前後測成對樣本 t 檢定分析結果

測驗	人數	平均分數	標準差	t 值
前測	34	42.35	18.43	-6.437***
後測	34	69.71	24.31	

****p<.001*

4.2. 有無完成學習單之學生學習成效比較

有完成學習單的學生共有 25 位，其後測成績(M=76.40; SD=22.89)顯著高於沒有完成學習單的學生共有 9 位，其後測成績(M=43.33; SD=26.93)，透過曼惠尼無母數分析結果，達顯著差異(Z=-2.99, p=0.003)。結果顯示在課程中有完成 WSQ 學習單的學生其學習成效高於在課程中沒有完成 WSQ 學習單的學生。統計結果如下表 2。

表 2 曼惠妮無母數分析結果

學習單	人數	平均分數	標準差	平均等級	等級總和	U 值	Z 值	*p*
完成	25	76.40	22.89	20.52	513.00	37.00	-2.99**	.003
未完成	9	43.33	26.93	9.11	82.00			

***p*<.01

5. 結論

本研究探討結合自主學習與混成式學習並且利用數位學習平台-因材網來提升 12 年級技術高中學生學習成效。研究結果顯示，透過四學理論並且搭配數位學習平台-因材網自主學習可以提高學生的學習成效，證明此學習模式能有效提升學生對於匯流排拓樸的理解與應用能力，同時，若有將 WSQ 學習單完成的學生，也會獲得更高的學習成效，進一步驗證了若在自主學習當中使用學習單作為學習輔助工具對於促進自主學習與知識深化上具有關鍵作用，和 Bernacki 等人（2020）研究有相同的結果，具備較強自主學習策略的學生能夠在缺乏即時教學支持的情境下仍維持良好的學習成果，Broadbent 和 Poon（2015）透過系統性文獻回顧指出，時間管理、後設認知與努力調節等自我調節學習策略，與線上學習成就之間具有顯著，顯示具備這些能力的學生在數位學習中較可能表現良好。

而在本研究所使用的數位學習平台-因材網，在臺灣教育部作為科技輔助自主學習的數位教學平台，能夠幫助學生透過個人化學習資源來強化自主學習能力，已在國小、國中與高中(職)階段廣泛應用，並且研究證實其能夠提升學生的學習動機與學習成效(Hung & Huang, 2022)，此外，教師在課堂中適時提供引導與回饋，對於學生的自主學習歷程具有關鍵影響 （Ho, 2022）。

本研究建議在未來研究中可將因材網當中所提供的 AI 學習夥伴-E 度整合至學習過程中，以提升學生的學習成效。根據 Chou et al.（2022）的研究，人工智慧技術在學習環境中的應用，透過人機互動提供個人化的學習體驗，並能根據學生的學習表現進行調整，幫助學生補充知

識盲點，提升學習興趣。因此，未來可以更著重探討因材網中 AI 學習夥伴-E 度是否可以更加提升學生的學習成效以及學習動機。

致謝

本研究感謝文冠甯助理協助因材網行政事宜，並且感謝部份國科會經費補助(計畫編號: NSTC 111-2410-H-003 -168 -MY3, 113-2628-H-003-002)。

參考文獻

Bernacki, M. L., Vosicka, L., & Utz, J. C. (2020). Can a brief, digital skill training intervention help undergraduates "learn to learn" and improve their STEM achievement?. *Journal of Educational Psychology*, *112*(4), 765.

Broadbent, J., & Poon, W. L. (2015). Self-regulated learning strategies & academic achievement in online higher education learning environments: A systematic review. *The internet and higher education*, *27*, 1-13.

Cao, W. (2023). A meta-analysis of effects of blended learning on performance, attitude, achievement, and engagement across different countries. *Frontiers in psychology*, *14*, 1212056.

Chou, C. M., Shen, T. C., Shen, T. C., & Shen, C. H. (2022). Influencing factors on students' learning effectiveness of AI-based technology application: Mediation variable of the human-computer interaction experience. *Education and Information Technologies*, *27*, 8723–8750. https://doi.org/10.1007/s10639-021-10866-9

Dakhi, O., Irfan, D., Jama, J., Ambiyar, A., Simatupang, W., Sukardi, S., & Zagoto, M. M. (2022). Blended learning and its implications for learning outcomes in computer and basic networks for vocational high school students in the era of COVID-19 pandemic. *International Journal of Health Sciences*, *6*(S4), 11177–11186.

Dziuban, C., Graham, C. R., Moskal, P., Norberg, A., & Sicilia, N. (2018). Blended learning: The new normal and emerging technologies. *International Journal of Educational Technology in Higher Education*, *15*(3).

Graham, C. R., & Halverson, L. R. (2022). Blended learning research and practice. In O. Zawacki-Richter & I. Jung (Eds.), *Handbook of open, distance and digital education* (pp. 1–17).

Ho, S. C. (2022). Curriculum practice of science and technology assisting primary school children's independent learning. *School Administrators*, (140), 103-121.

Hung, S. H., & Huang, J. W. (2022). A study on the use intention of elementary school students in adaptive learning platform based on technology acceptance model. *School Administrators*, (139), 114-133.

Jansen, R. S., Van Leeuwen, A., Janssen, J., Jak, S., & Kester, L. (2019). Self-regulated learning partially mediates the effect of self-regulated learning interventions on achievement in higher education: A meta-analysis. *Educational Research Review*, *28*, 100292.

Jumaini, H., Hertin, H. H., Nisfiyati, M., & Ibrahim, M. (2021). Penerapan Metode Pembelajaran Blended Learning dalam Meningkatkan Pemahaman Konsep Hasil Belajar Siswa: Sebuah Meta-Analisis. Al Khawarizmi: *Jurnal Pendidikan dan Pembelajaran Matematika*, *5*(1), 48-63.

Li, S., & Wang, W. (2022). Effect of blended learning on student performance in K-12 settings: A meta-analysis. *Journal of Computer Assisted Learning*, *38*(5), 1254–1272. https://doi.org/10.1111/jcal.12696

Liu, X. (2024). The role and impact of online learning platforms in higher education. *Adult and Higher Education*, *6*(5), 1–8. https://doi.org/10.23977/aduhe.2024.060501

Nácher, M. J., Badenes-Ribera, L., Torrijos, C., Ballesteros, M. A., & Cebadera, E. (2021). The effectiveness of the GoKoan e-learning platform in improving university students' academic performance. *Studies in Educational Evaluation*, *70*, 101026.

Ordones, S. A. D., & Ferneda, E. (2022). The contribution of digital learning platforms to undergraduate teaching. *Revista EDICIC*, *2*(2), 1–11.

Rakic, S., Tasic, N., Marjanovic, U., Softic, S., Lüftenegger, E., & Turcin, I. (2020). Student performance on an e-learning platform: Mixed method approach. *International Journal of Emerging Technologies in Learning (iJET)*, *15*(2), 187–203.

Schunk, D. H., & Zimmerman, B. J. (1994). *Self-regulation of learning and performance: Issues and educational applications*. Lawrence Erlbaum Associates Inc.

Tuilan, J. (2023). Self-regulated learning In blended learning environment. Script Journal: *Journal of Linguistics and English Teaching*, *8*(2), 154-162.

Vosniadou, S. (2020). Bridging secondary and higher education. The importance of self-regulated learning. *European Review*, *28*(S1), S94-S103.

Xie, H., & Wei, L. (2020). Design of networked and digital teaching platform based on big data. Journal of Physics: *Conference Series*, *1575*(1), 012120.

Zhao, M., Liao, H.-T., & Sun, S.-P. (2020). An education literature review on digitization, digitalization, datafication, and digital transformation. Advances in Social Science, *Education and Humanities Research*, *435*, 301–305.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Exploring Primary School Teachers' Perspectives in Integrating AI into STEM Education Through Modular STEM Activities

Pui Yiu TAM[1*], Muhammad ALI[2], Gary K.W. WONG[3]
[1,2,3]The University of Hong Kong, Hong Kong SAR
u3573058@connect.hku.hk, akula@connect.hku.hk, wongkwg@hku.hk

Abstract: *This study explores the integration of AI education into a Hong Kong primary school through modular STEM activities, addressing challenges related to teacher preparedness and curriculum design. Collaborating with three STEM teachers, we co-developed hands-on learning modules delivered via an after-school club. Qualitative analysis of observations, interviews, and learning artifacts revealed teachers' initial conceptual and technical struggles, which were mitigated by iterative co-design and low-code tools. While modular resources lowered entry barriers, dependency risks emerged, particularly among novice educators relying on predefined learning content. Experienced teachers demonstrated adaptive innovation, repurposing STEM pedagogies for AI contexts. Findings emphasize the dual role of modular STEM activities—standardizing implementation while enabling flexibility for skilled educators. Furthermore, the findings underscore the need for structured teacher professional development (TPD) that provides scaffolded support to promote autonomy among in-service teachers, fostering critical AI literacy and pedagogical agency.*

Keywords: AI education, STEM activities, teacher professional development, co-design, modular learning

1. Introduction

Educational stakeholders are collaborating to incorporate AI education into K–12 curricula, preparing the next generation for a future shaped by rapidly advancing AI, with early exposure helping to establish a solid foundation for future learning (Walter, 2024). Following Hong Kong's prior experience in developing an AI curriculum for secondary schools (Chiu et al., 2022), this study serves as a pilot project focusing on age-appropriate AI education at the primary school level to lay the groundwork for earlier and more comprehensive AI literacy. Dai et al. (2023) demonstrated that primary school teachers can create an inclusive and accessible AI curriculum by embedding it within an existing STEM framework while ensuring academic rigor. Therefore, we argue that active STEM exploration in Hong Kong primary schools (Wong, 2017) provides a viable pathway for incorporating AI into the curriculum. For instance, despite the complexity of AI, independent learning modules covering AI concepts, applications, and ethics can be developed to allow for flexible implementation and easy entry for beginners (Kandlhofer et al., 2023). However, integrating AI into educational settings presents challenges, particularly due to teachers' limited preparedness in this emerging arena. Ng et al. (2022) argue that the rapid development of AI-based tools by technology companies—often without sufficient consideration of teachers' and students' needs—may further overwhelm educators. To bridge this burgeoning gap, we engaged in iterative co-design with three Hong Kong primary STEM teachers, implementing a series of modular STEM activities while tracking teachers' perspectives in integrating AI into STEM education throughout the process.

2. Methods

2.1. Context and Participants

This study was conducted in a private co-educational primary school in Hong Kong—well-equipped with STEM learning resources—focusing on three teachers (see Table 1 for background information) who collaborated with our research team to design and implement a series of modular STEM activities centered on AI technologies. These activities were delivered through an after-school STEM club, comprising 17 mixed-age students from Primary 4 to Primary 6. The students were selected for their strong interest and aptitude in STEM, as they frequently engaged in innovative projects for STEM competitions under their teachers' guidance.

2.2. Collaboration between the research team and the teachers

To initiate the collaboration, the research team designed a preliminary framework for modular STEM activities, integrating key aspects of STEM education, such as interdisciplinarity, constructionist learning, and real-world problem-solving (Ali, 2021; Ali et al., 2024). This framework was developed as a robust instructional package, including teacher guidelines, student workbooks, and activity slides. To ensure effective co-design, we first supported teachers in gaining a broad overview of AI by providing appropriate resources and helping them create tailored learning content. Training sessions were then conducted to guide them through each learning module while offering relevant background information. Once they became familiar with the content, they assessed whether the planned activities aligned with students' cognitive levels, refining or devising new activities as needed. Their feedback informed formative adjustments to the learning modules, ensuring their relevance and effectiveness. As the implementation progressed, the teachers engaged in regular discussions with us, reviewing and refining the content before each session. This iterative process of training, feedback, and continuous refinement aimed to facilitate the seamless integration of AI education into the classroom.

Table 1. Background information of three teachers

Name (Pseudonym)	Scott	Winnie	Yvette
Gender	M	F	F
Age Range	40 - 45	25 - 30	20 - 25
Teaching Subjects	Science Studies, General Studies	Math, Science Studies, Computer Studies	Math, Computer Studies
Years of School Subject-teaching	17	5	2

2.3. Final Learning Modules

Through the aforementioned collaboration, the final series of activities was implemented as five learning modules across eight once-weekly sessions, each lasting approximately 50 minutes. These modules included: (1) "*What is AI?*", introducing AI through discussions and interactions with a virtual assistant (e.g., Siri); (2) "*What is Machine Learning?*", using Google's Quick, Draw! to demonstrate supervised learning through pattern recognition; (3) "*Utilizing Machine Learning in Daily Problem-Solving*", leveraging Google Teachable Machine (GTM) to train an image recognition model applied to a Micro-Bit for a smart waste bin prototype; (4) "*How Classifiers Work: Supervised and Unsupervised Learning*", contrasting supervised and unsupervised methods through card-sorting activities; and (5) "*Exploring GenAI and Its Risks*", analyzing ethical concerns by comparing human- and AI-generated content (e.g., Midjourney, Sora). Each module prioritized hands-on engagement to connect AI concepts to real-world contexts.

2.4. Data Source and Analysis

We conducted an exploratory study to gain insights into the perspectives of participating teachers in integrating AI into their classrooms through modular STEM activities. Multiple qualitative data sources, collected across all stages of the collaboration, were used for triangulation and to create a comprehensive description. These included: (a) participant observations (e.g., field notes, video recordings) from AI workshops, project meetings, and activity implementations; (b) teaching artifacts (e.g., revised activity slides and student workbooks); and (c) post-implementation interviews, during which each of the three teachers was individually interviewed for 45–60 minutes using semi-structured interview questions. To analyze this extensive data, an interpretive approach was employed (Merriam, 1998). Thematic analysis (Braun & Clarke, 2012), noted for its flexibility, was used to identify key meaning patterns and capture the major perspectives teachers demonstrated as the collaboration progressed. Narrative analysis (Fina et al., 2021) complemented this by preserving continuity and contradictions in teachers' experiences, offering a holistic understanding of their thematic interpretations.

3. Findings

This section presents selected findings on the participating teachers' perspectives. The thematic analysis identified *three* overarching themes that aligned with their predominant narratives, later confirmed through narrative analysis.

3.1. Challenges in Teachers' Conceptual Understanding and Technical Adaptability

The implementation of AI education revealed interconnected challenges in the teachers' conceptual understanding and technical adaptability. Initially, knowledge-related anxiety stemmed from AI's disciplinary complexity, with experienced educators like Scott expressing concern: "*As I can't fully grasp the concept of AI, I worry about providing students with an incorrect entry point*…" (Preparatory Meeting). He also noted that the school's prior attempts to introduce AI instruction had been hampered by teaching materials that prioritized tool-specific procedures over conceptual explanations, forcing teachers to rely on step-by-step tutorials without fostering deeper understanding.

Targeted training sessions addressed these gaps, prompting critical reflection—for instance, Scott recognized that traffic light tasks, with their fixed outcomes, poorly modelled machine learning's predictive logic (Training Session). Classroom practice itself became a catalyst for growth: Winnie, a less experienced teacher, noted how teaching AI deepened her grasp of concepts like supervised learning, while novice educator Yvette acknowledged that "*reviewing the material myself* [while teaching] *deepened my understanding*" (Interview). Yet systemic constraints persisted. The teachers' expertise remained bound to module content, with Winnie admitting her understanding was "*limited to the slides I teach*" due to the absence of formal AI integration (Interview). This dependency hindered instructional autonomy, as Scott conceded: "*We rely heavily on your* [researchers'] *guidance… It's your ideas we implement*" (Interview).

These conceptual uncertainties amplified technical struggles during the implementation. The teachers reported that programming complexity—evident in prior STEM lessons derailed by Python coding errors (Winnie, Preparatory Meeting)—prompted choices of low-code tools like GTM and Micro-Bit. While these tools enabled tangible projects (e.g., AI-powered smart trash bins) and reduced syntax barriers, they risked oversimplification. Scott critiqued GTM's narrow utility, stating, "*The model can't distinguish glass from plastics, limiting real-world problem-solving*" (Progress Meeting), while interdisciplinary knowledge gaps hindered deeper task elaboration. Attempts to leverage GenAI (e.g., GPT-4o) for code generation failed due to prompt specificity demands, revealing persistent reliance on external expertise.

3.2. Teachers' Adaptation of AI Instruction Through Existing Pedagogical Expertise

Leveraging their experience in project-based and inquiry-based STEM teaching, the teachers intuitively adopted constructionist approaches to co-design the learning modules, despite their limited AI expertise. They applied these approaches in AI contexts, for example, by integrating interactions with Siri—a virtual assistant that students were familiar with—to highlight AI's prevalence and key features, such as natural interaction, through firsthand experience. Most notably, post-implementation reflections reinforced the value of hands-on, experiential learning. Scott noted that tools like GTM boosted engagement by letting students "*experiment with real-life scenarios*" (Interview), while Yvette emphasized how Quick, Draw! demystified machine learning through direct interaction with input-output cycles: "*We* *should find more tools like Quick, Draw! for students to experience… Letting them further explore the underlying mechanisms*" (Interview).

Furthermore, the teachers demonstrated their capacity for proactive problem-solving in adapting AI tools. For example, when students became confused while training the image recognition model on GTM, Scott identified the source of the confusion: non-standardized target objects (i.e., using various types of waste bags, as seen in Figure 1, left). He then decided to standardize the task using specific waste bags from an MSW charging scheme (Figure 1, right) as the training objects, enhancing the task authenticity. Due to the scarcity of these bags at the time, Scott visited multiple convenience stores to procure them and shared this effort with the students during the lesson: "*I ran to several convenience stores for the bags*" (Lesson Observation). As a result, the students better understood the task setting and successfully developed the corresponding AI model.

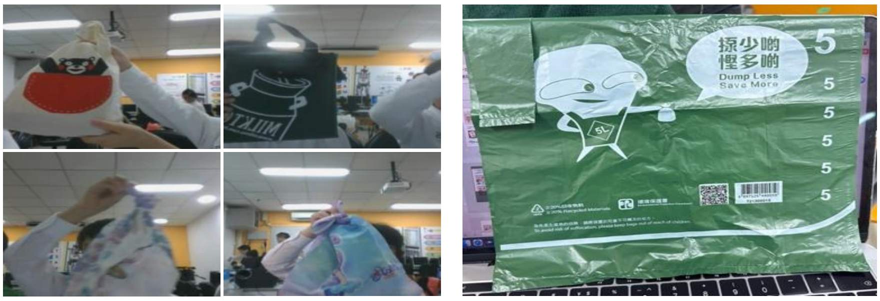


Figure 1. Non-standard target objects (left) and a standard designated bag as the target object (right).

3.3. The Dual Role of Modular Design in AI Education Adoption

Initially, the teachers exhibited a high degree of dependency on the research team for activity design, preferring that the team provide complete teaching content. Accordingly, the structured guidance of the modules (including step-by-step procedures and visual aids) lowered the implementation threshold. Winnie noted that the resources clarified uncertainties: "*We can identify the problems* [within the resources] *and instantly ask you if unclear*" (Interview). Scott advocated for the necessity of modularity for scalability: "*It enables teachers to start first, improve through trial and error… critical for primary schools*" (Interview). However, adoption differed among the teachers. Scott, the most experienced teacher, proactively customized the activities based on his previous STEM teaching experience. Meanwhile, the two younger teachers, Winnie and Yvette, preferred following the original design and focused on the integrity of delivering the content. Scott observed: "*The other two teachers are younger, still being trained in our atmosphere to complete tasks. But when I am a bit older, I would think that if some things can be cut and are not that important, I'd rather spend more time on essential aspects*" (Interview). These differences suggest that professional stage can affect AI integration. The experienced teacher demonstrated adaptable innovativeness, while the younger teachers relied on the well-structured content to avoid the risk of uncertainty. This divergence highlights the dual role of modular design—standardizing implementation for broad adoption while enabling adaptable educators to innovate within flexible frameworks.

4. Discussion

4.1. Scaffolding AI Education: Bridging Tools, Dependency and Autonomy

Modular instructional design and low-code AI tools collectively function as dual scaffolding mechanisms, lowering barriers to AI education adoption. While modular resources enable teachers to bypass content-reconstruction challenges and focus on pedagogical execution, low-code tools simplify technical implementation through drag-and-drop interfaces (Kim & Kwon, 2024). However, both approaches risk perpetuating dependency: teachers remain reliant on predefined modules for lesson design and prepackaged tool workflows that obscure algorithmic transparency, exemplifying the limitations of over-scaffolding—where rigid structures inhibit autonomous conceptual exploration.

To foster sustained autonomy, teacher professional development (TPD) for AI education must strategically balance structured support (e.g., modular instructional designs) to reduce entry barriers with adaptive competency-building. This dual approach addresses AI's inherent complexity while empowering educators to innovate beyond prescriptive frameworks. Future research should refine the AI-integrated Technological Pedagogical Content Knowledge (TPACK) model (Ng et al., 2022; Kim et al., 2021), bridging pre-service training in higher education with in-service TPD to position educators as intentional designers of AI pedagogy.

For technological knowledge (TK) (Koehler & Mishra, 2005), future designs should integrate explanatory scaffolding—layered interfaces that progressively reveal AI principles (e.g., CNN Explainer), enabling learners to transition from basic algorithmic logic to complex applications. Additionally, with GenAI's rapid development, educators and learners can leverage personalized programming assistants to interpret complex concepts and debug code (Boguslawski et al., 2025). Notably, effectively utilizing these tools requires foundational proficiency in GenAI applications. By strengthening TK, educators can demystify AI's ‘black box’ in classrooms, ensuring sustained technical accessibility.

4.2. PCK-Driven Collaborations among Differentiated Educators for Effective TPD

This study observed divergent trajectories in AI instructional design adoption among three teachers, shaped by their professional development stage. The experienced educator demonstrated agency in innovating module content, suggesting that structured modules can act as catalysts for teacher-led innovation rather than rigid scripts. In contrast, novice teachers adhered closely to pre-set workflows, prioritizing fidelity to mitigate risks associated with unfamiliar AI pedagogies—a behavior consistent with early-career teachers' reliance on external guidance to navigate complexity (Berliner, 2004). Notably, even the most novice teacher contributed ideas for integrating individual innovations to refine the design. As the teachers' pedagogical content knowledge (PCK) surpassed the modules' preset boundaries, this finding echoes Desimone's (2009) assertion that PCK maturation enables educators to adapt standardized resources to context-specific needs. To leverage such dynamics, we propose differentiated collaborative strategies for TPD. First, scaffold early-career teachers through cognitive apprenticeships (Kolikant et al., 2006), pairing them with seasoned peers to co-design modules, thereby reducing anxiety while fostering creative agency. Second, institutionalize cross-phase communities of practice (Lave & Wenger, 1991), where veteran teachers model iterative design processes, demystifying innovation for novices.

Nevertheless, this study reveals that modular STEM activities provide primary educators with accessible entry points for AI instruction. However, sustained integration requires structured TPD that extends beyond basic adoption. While teachers demonstrated adaptive reuse of STEM pedagogy in AI contexts, effective curriculum integration necessitates ongoing support to ensure age-appropriate learning. Although initial success largely stems from educators' prior STEM expertise, long-term efficacy depends on TPD aligned with TPACK, fostering autonomous pedagogical innovation rather than passive compliance. Crucially, TPD must also prioritize educators' critical evaluation of AI tools, particularly the epistemic assumptions of these technologies and their alignment with the constructionist principles of STEM education.

References

Ali, M. (2021). State of STEM Education in Hong Kong: A Policy Review. *Academia Letters*, Article 3680. https://doi.org/10.20935/AL3680

Ali, M., Wong, G. K.-W., & Ma, M. (2024). K–12 Pre-service Teachers' Perspectives on AI Models and Computational Thinking: The Insights from an Interpretative Research Inquiry. *Proceedings of the 8th APSCE International Conference on Computational Thinking and STEM Education (CTE-STEM 2024), 8*(1), 66-71. https://doi.org/10.5281/zenodo.11559685

Braun, V., & Clarke, V. (2012). Thematic analysis. In H. Cooper, P. M. Camic, D. L. Long, A. T. Panter, D. Rindskopf, & K. J. Sher (Eds.), *APA handbook of research methods in psychology, Vol 2: Research designs: Quantitative,* *qualitative, neuropsychological, and biological.* (pp. 57–71). American Psychological Association. https://doi.org/10.1037/13620-004

Berliner, D. C. (2004). Describing the Behavior and Documenting the Accomplishments of Expert Teachers. *Bulletin of Science, Technology & Society, 24*(3), 200–212. https://doi.org/10.1177/0270467604265535

Boguslawski, S., Deer, R., & Dawson, M. G. (2025). Programming education and learner motivation in the age of generative AI: Student and educator perspectives. *Information and Learning Sciences*, *126(*1/2), 91–109. https://doi.org/10.1108/ILS-10-2023-0163

Chiu, T. K. F., Meng, H., Chai, C.-S., King, I., Wong, S., & Yam, Y. (2022). Creation and Evaluation of a Pretertiary Artificial Intelligence (AI) Curriculum. *IEEE Transactions on Education*, *65*(1), 30–39. https://doi.org/10.1109/TE.2021.3085878

Dai, Y., Liu, A., Qin, J., Guo, Y., Jong, M. S., Chai, C., & Lin, Z. (2023). Collaborative construction of artificial intelligence curriculum in primary schools. *Journal of Engineering Education, 112*(1), 23–42. https://doi.org/10.1002/jee.20503

De Fina, A. (2021). Doing narrative analysis from a narratives-as-practices perspective. *Narrative Inquiry, 31*(1), 49–71. https://doi.org/10.1075/ni.20067.def

Desimone, L. M. (2009). Improving Impact Studies of Teachers' Professional Development: Toward Better Conceptualizations and Measures. *Educational Researcher*, *38*(3), 181–199. https://doi.org/10.3102/0013189X08331140

Kandlhofer, M., Weixelbraun, P., Menzinger, M., Steinbauer-Wagner, G., & Kemenesi, Á. (2023). Education and Awareness for Artificial Intelligence. In J.-P. Pellet & G. Parriaux (Eds.), *Informatics in Schools. Beyond Bits and Bytes: Nurturing Informatics Intelligence in Education* (Vol. 14296, pp. 3–12). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-44900-0_1

Kim, K., & Kwon, K. (2024). Tangible computing tools in AI education: Approach to improve elementary students' knowledge, perception, and behavioral intention towards AI. *Education and Information Technologies*, *29*(13), 16125–16156. https://doi.org/10.1007/s10639-024-12497-2

Kim, S., Jang, Y., Choi, S., Kim, W., Jung, H., Kim, S., & Kim, H. (2021). Analyzing Teacher Competency with TPACK for K-12 AI Education. *KI - Künstliche Intelligenz*, *35*(2), 139–151. https://doi.org/10.1007/s13218-021-00731-9

Koehler, M. J., & Mishra, P. (2005). Teachers Learning Technology by Design. *Journal of Computing in Teacher Education*, *21*(3), 94–102. https://doi.org/10.1080/10402454.2005.10784518

Kolikant, Y. B.-D., Gatchell, D. W., Hirsch, P. L., & Linsenmeier, R. A. (2006). A Cognitive-Apprenticeship-Inspired Instructional Approach for Teaching Scientific Writing and Reading. *Journal of College Science Teaching, 36*(3), 20–25.

Lave, J. (with Wenger, E.). (1991). *Situated learning: Legitimate peripheral participation*. Cambridge University Press.

Merriam, S. B. (1998). *Qualitative Research and Case Study Applications in Education—Revised and Expanded from "Case Study Research in Education."* Jossey-Bass Publishers, 350 Sansome St, San Francisco, CA 94104.

Ng, D. T. K., Leung, J. K. L., Su, M. J., Yim, I. H. Y., Qiao, M. S., & Chu, S. K. W. (2022). *AI Literacy in K-16 Classrooms*. Springer International Publishing. https://doi.org/10.1007/978-3-031-18880-0

Walter, Y. (2024). Embracing the future of Artificial Intelligence in the classroom: The relevance of AI literacy, prompt engineering, and critical thinking in modern education. *International Journal of Educational Technology in Higher Education*, *21*(1), 15. https://doi.org/10.1186/s41239-024-00448-3

Wong, G. K. W. (2017). Integrative learning in K-12 STEM education: How to prepare the first step? *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 80–87. https://doi.org/10.1109/TALE.2017.8252308

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Exploring Teacher Beliefs about Teaching AI Ethics Under National Curriculum Reform: A Theory of Planned Behavior Perspective

Ming MA[1*], Davy Tsz Kit NG[2], Gary K.W. WONG[3]
[1,3]The University of Hong Kong, Hong Kong SAR
[2]The Education University of Hong Kong, Hong Kong SAR
mingma@connect.hku.hk, davyngtk@eduhk.hk, wongkwg@hku.hk

Abstract: *This study investigates secondary school teachers' beliefs about teaching AI ethics within the context of China's national curriculum reform. Grounded in the Theory of Planned Behavior (TPB), the research analyzes semi-structured interviews with 14 in-service technology teachers experienced in delivering AI curricula. Findings reveal teachers' beliefs across three TPB domains: 1) behavioral beliefs (perceived benefits and challenges of teaching AI ethics), 2) normative beliefs (social and institutional expectations), and 3) control beliefs (perceived self-efficacy and resource availability). These insights highlight the complex interplay between personal, social, and structural factors shaping AI ethics education. The study contributes to the growing discourse on AI literacy by offering evidence-based recommendations for teacher professional development and curriculum design, ultimately supporting effective AI education integration in schools.*

Keywords: AI ethics education, Theory of Planned Behavior (TPB), Teacher Beliefs, Curriculum Reform

1. Introduction

AI education is increasingly being integrated into K-12 curricula worldwide, driven by the global push for AI literacy, which reflects the growing recognition of AI's curricula role in shaping future human society (Miao & Shiohira, 2022; Ng et al., 2023). Central to many well-established AI literacy frameworks is the emphasis on AI ethics, referring to the competence of effectively engaging with AI in an ethical and responsible manner (e.g., Kong et al., 2024; Long & Magerko, 2020; Stolpe & Hallström, 2024; Wong et al., 2020). However, the teaching of AI ethics, which is essential for K-12 students to develop their AI literacy, remains largely underprioritized among educators. According to UNESCO report of government-endorsed AI curricular report, AI ethics is the part with high commitment but least implementation in the K-12 classroom (Miao & Shiohira, 2022). Particularly in China, while the government mandated that high school students (Grades 10-12) begin learning AI in 2018 and introduced a new compulsory National Information Science & Technology Curriculum Standard (Grades K-9) in 2022, with AI ethics as a key learning area, a recent analysis of teachers' classroom instructions revealed that AI ethics accounts for only 5.1% of overall AI classroom instruction, highlighting a significant practical gap in its practical implementation (Wu et al., 2024).

Research on teachers' cognitions has consistently demonstrated that teachers' beliefs play as central roles in shaping their classroom behaviors, such as acting as filters, interpretive devices, and transformers of externally developed curricular intentions (Borko & Putnam, 1996; Kagan, 1990; Pajares, 1992), These beliefs significantly impact the success of curriculum reforms across different subjects and cultural contexts (Bryan, 2012; Ham & Dekkers, 2019; Handal & Herrington, 2003). However, existing literature on K-12 AI education lacks insights into teachers' beliefs, particularly in the context of national curriculum reform and their perspectives on teaching AI ethics. Although a few studies have explored teachers' conceptions or perspectives of overall AI or ML education (e.g., Williams et al., 2021; Yau et al., 2023), AI ethics were only briefly mentioned as an essential component, without in-depth investigation into how they are integrated or taught. Moreover, most of these studies have been conducted within the context co-designed AI initiatives with researchers, rather than through top-down curriculum implementation. Therefore, this study aims to understand teachers' beliefs and practices in teaching AI ethics within the context of the national curriculum reform in China.

2. Theoretical Framework

We employed the Theory of Planned Behavior (TPB) (Ajzen, 1985, 1991), a prominent framework that has been widely used in previous literature to study teachers' beliefs and how these beliefs influence their professional behaviors across various subjects (e.g., Urton et al., 2023; Voet & De Wever, 2020). TPB provides a structural overview of teachers' beliefs in three dimensions: behavioral beliefs, normative beliefs, and control beliefs (See Figure 1). These belief-based measures serve as the foundational antecedents to the core constructs of TPB—attitudes, subjective norms, and perceived behavioral control—and offer a nuanced understanding of the cognitive and social factors that shape teachers' intentions and behaviors. Behavioral beliefs refer to teachers' perceptions about the likely outcomes of performing a specific behavior and the evaluations of those outcomes, which translate into attitudes toward the behavior. Normative beliefs reflect teachers' perceptions of the social pressures to perform or not perform a behavior, shaped by the expectations of significant others, such as colleagues, school administrators, parents, and educational policymakers. Control beliefs refer to teachers' perceived ability to perform a behavior, encompassing their perceptions of both internal factors (e.g., self-efficacy, knowledge, and skills) and external factors (e.g., resources, institutional support, and time constraints) that may facilitate or hinder their actions. This study investigates behavioral, normative, and control beliefs to identify the underlying factors shaping teachers' attitudes, subjective norms, and perceived behavioral control in implementing AI ethics education. Specifically, the paper addresses the following research question: What key factors influence teachers' beliefs about teaching AI ethics?

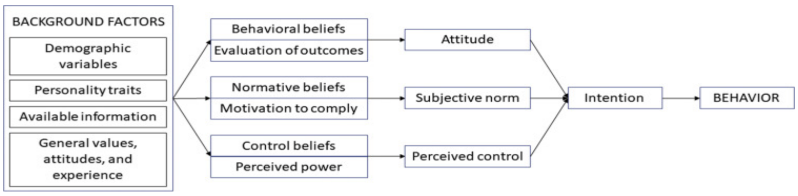


Figure 1. Theory of Planned Behavior, modified from Ajzen (1985, 1991)

3. Methods

3.1. Participants and contexts

Qualitative interviews were conducted with 14 secondary technology teachers. These teachers are in-service teachers of information technology, in public secondary schools, of the great bay areas in China.

3.2. Procedure

Semi-structured interviews (15–50 minutes, M = 32 minutes) explored teachers' pedagogical experiences of teaching AI ethics. Interviews continued until data saturation, conducted via videoconference, audio-recorded, and transcribed verbatim. The transcripts were initially in Chinese and then translated into English for analysis. Ethical protocols included informed consent and anonymization.

3.3. Data analysis

Data were analyzed using thematic analysis based on Braun and Clarke's (2006) framework, combining both deductive and inductive coding approaches. The analysis was guided by the Theory of Planned Behavior (TPB), with the initial coding scheme derived from the three core components of TPB: Behavioral Beliefs, Normative Beliefs, and Control

Beliefs. These predefined categories were used to analyze teachers' beliefs regarding AI ethics education. In addition to the deductive coding, an inductive coding process allowed for the emergence of new themes that were not explicitly captured by the TPB framework.

4. Results

4.1. Behavioral beliefs

The Behavioral Beliefs category explored teachers' perceptions of the benefits of teaching AI ethics, with a strong emphasis on its importance in preparing students for an AI-driven future. Teachers consistently recognized AI ethics as a crucial component of the curriculum. As Teacher 2 stated, "It (AI ethics) is very important after adding this course because nowadays this artificial intelligence along with this. Computer. The development of computational power. Unlike some of the past, Semi-automated artificial intelligence, now the artificial intelligence, it's this. Intelligent degrees are very high, it can be applied in many emerging fields. For example, the current is very hot. Autopilot, driverless, which involves many ethical issues within that." This reflects the perceived importance of AI ethics integration in the IT curriculum, highlighting how teachers value AI ethics as foundational for students' understanding of the societal impact of AI. While teachers acknowledged the importance of teaching AI ethics, they also identified significant challenges in doing so. The primary challenge was the complexity of AI ethics, with teachers highlighting the difficulty of addressing abstract and multifaceted issues that need to be addressed. As Teacher 6 noted, "AI ethics is a complex subject... it involves many issues, like data privacy and the attribution of responsibility." Another major challenge mentioned was the difficulty of keeping up with the rapid development of AI technology. Teachers expressed concern about the need to maintain the currency and relevance of the content in a field that evolves so quickly. As one teacher observed, "Maintaining the currency and relevance of the content is challenging due to rapid AI development. Ethical issues evolve with the technology."

Many teachers also discussed the moral responsibility of students when engaging with AI technologies. For example, Teacher 5 remarked, "Artificial intelligence has both pros and cons. For example, it may involve data awareness... students need to understand how to use it responsibly." This statement aligns with the moral development aspect of behavioral beliefs, where teachers emphasized the necessity for students to not only understand the technical aspects of AI but also its ethical implications, fostering responsible usage of technology. Furthermore, teachers highlighted the real-world relevance of AI ethics, noting that AI is already integrated into daily life through technologies like smartphones and voice assistants. As Teacher 9 pointed out, "AI is already in our lives, like in smartphones and voice assistants, so students need to learn the ethical implications of these technologies." This statement reinforces the notion that teaching AI ethics is essential for helping students understand the practical applications of AI and the ethical dilemmas associated with its use in everyday settings. Additionally, a few teachers also noted that AI ethics aligned with their personal values of education. or example, Teacher 12, who also serves as the school's chancellor, emphasized that the ultimate goal of his educational philosophy is to nurture well-rounded individuals who are not only technical experts but also responsible and engaged citizens.

4.2. Normative beliefs

The Normative Beliefs category reflects the external influences that shape teachers' perceptions of the importance and necessity of teaching AI ethics, including peer expectations, school leadership, and curriculum policies. First, teachers frequently discussed how their school's stance on AI ethics shaped their own teaching priorities. As Teacher 4 noted, "If the school management recognizes the importance of AI, there will definitely be pressure on us to teach it." Additionally, Teacher 8 reported that although local educational authorities aim to push AI education in IT classrooms, many schools have yet to implement it, primarily due to a lack of attention and insufficient qualified teachers. This highlights the significant role of school leadership in either supporting or pressuring teachers to integrate AI ethics into their curriculum.

Another significant external influence was the lack of peer collaboration on teaching AI ethics. While teachers acknowledged the importance of the subject, opportunities for peer discussions and collaboration on effective teaching strategies for AI ethics were limited. As Teacher 5 observed, “We talked about AI ethics once or twice, but no real collaboration has happened yet.” This suggests that, although teachers recognize the importance of AI ethics, peer support for developing teaching strategies remains minimal, likely due to a lack of formal structures or institutional guidance on this topic. Despite limited peer collaboration, teachers generally agreed on the importance of teaching AI ethics. As Teacher 13 stated, “In my discussions with other teachers, there's a general consensus that AI is an important development. It should be included in teaching. Recognize and support!” This statement highlights a shared understanding among teachers of the subject's significance but also underscores the lack of formalized support and collaboration in developing strategies to teach AI ethics effectively.

Additionally, the perceived interest of students in AI ethics also influenced teachers' attitudes towards teaching the subject. Teachers observed that students, particularly those who are already familiar with AI technologies like smartphones and social media, expressed increasing curiosity about the ethical issues surrounding AI. As Teacher 1 stated, “Students are really curious about the consequences of AI, especially when we talk about issues like privacy and responsibility in AI systems.” This perceived student interest adds an external layer of pressure for teachers to include AI ethics in their curriculum, as students' curiosity is growing, potentially influencing teachers' decisions to address the topic. Finally, external curriculum reforms and policies also shaped teachers' beliefs. While some teachers were optimistic about future developments in the curriculum, others expressed frustration with the lack of timely updates of the documents. Teacher 6 mentioned, “The curriculum standards are not always up to date with the latest AI developments, so we have to make adjustments on our own.” This points to a gap between official educational policies and the real-world pace of AI technology, which can make it difficult for teachers to stay aligned with current developments and ensure that their teaching remains relevant.

4.3. Control beliefs

The Control Beliefs category reflects teachers’ perceptions of the resources, constraints, and institutional support that influence their ability to effectively teach AI ethics. Teachers frequently discussed the challenges they faced in terms of available resources, personal expertise, and institutional support, which shaped their perceptions of the feasibility of teaching the subject. Specifically, Teachers identified significant barriers to teaching AI ethics due to limited hardware and software tools. Teacher 6 shared, “The (AI-based) hardware is still more restrictive... we cannot do that for all classes.” This statement highlights the lack of technological resources necessary to implement practical, hands-on lessons in AI ethics. While some teachers managed to conduct basic AI experiments with limited resources, the absence of more advanced tools hindered the depth and breadth of their lessons on AI ethics.

Another significant issue raised by teachers was their lack of personal expertise and the need for professional development in AI ethics. Many teachers acknowledged that they had not received formal training on the topic and were self-learning in order to teach AI ethics to their students. Teacher 4 noted, “If you're not familiar with it, you don’t know how to approach it... we need more training.” Similarly, teacher 2 also expressed that “I don't have a systematic way to learn these AI techniques, and he requires a very wide range of knowledge. Knowledge is very broad. I did not. In-depth to specialize in this knowledge (AI ethics) just now. Just from a very shallow surface to communicate with students...”. These statements reflect a strong need for professional development opportunities, which teachers felt were essential for their ability to teach AI ethics effectively but were often lacking or insufficient.

Teachers also pointed to institutional support as a critical factor in their ability to teach AI ethics. While some schools offered general support and teaching resources, others were less proactive in assisting teachers with implementing AI ethics in the curriculum. Teacher 5 mentioned, "If the school has that in mind and then has these training materials... then I think it will have a big impact." This emphasizes the importance of institutional backing, including training materials and structured guidance, in enabling teachers to teach AI ethics effectively. Additionally, classroom constraints such as time limitations and an overloaded curriculum were highlighted as significant barriers. Many teachers felt that the fast-paced curriculum left little room for in-depth discussions on AI ethics. As Teacher 8 explained, "With the packed curriculum, we don't have enough time to go in-depth into AI ethics." This reflects the challenge of fitting AI ethics into an already crowded schedule, making it difficult for teachers to dedicate sufficient time and attention to the subject.

5. Discussion

This study highlights several key challenges faced by teachers in integrating AI ethics into the curriculum. While a few teachers have experience teaching AI ethics, the majority expressed a lack of confidence due to gaps in their knowledge and the lack of adequate resources, such as textbooks and access to AI applications. These challenges, along with the need for greater peer support and collaboration, point to significant barriers in the effective teaching of AI ethics, presented as below:

1. Lack of Knowledge and Confidence: A common theme among teachers was the lack of expertise in AI ethics, which directly impacted their confidence in teaching the subject. Many teachers reported feeling unprepared to tackle the complex ethical issues associated with AI, such as data privacy and responsibility attribution. Without sufficient training or professional development, teachers found it difficult to engage with these abstract and multifaceted topics.
2. Resource Constraints: Teachers also highlighted the lack of resources available to support the teaching of AI ethics. While some had access to basic materials or local textbooks, these were often outdated or insufficiently detailed for teaching AI ethics effectively. The absence of AI applications and hands-on tools was a significant barrier, as teachers were unable to fully demonstrate the practical implications of AI ethics in their classrooms.
3. Need for Peer Support and Collaboration: Another prominent challenge was the lack of collaboration among teachers. While teachers acknowledged the importance of AI ethics, they reported limited peer support and collaborative opportunities for discussing and teaching the subject. This finding indicates that peer networks could play a critical role in enhancing teachers' confidence and sharing teaching strategies for AI ethics.

6. Implications

The findings underscore the need for targeted professional development to address the knowledge gaps teachers face in teaching AI ethics. Schools and educational authorities must provide comprehensive training to ensure teachers feel confident in their ability to engage students with complex ethical issues related to AI. Additionally, schools must prioritize the provision of updated resources, including textbooks and access to AI tools, to support teachers in implementing AI ethics lessons effectively. Furthermore, fostering peer support and creating collaborative opportunities for teachers to exchange knowledge and strategies will be crucial in overcoming the isolation many teachers feel when teaching AI ethics. Formal communities of practice or teacher networks could help build a supportive environment where teachers can share resources and discuss best practices.

Future studies could delve deeper into how peer collaboration and the establishment of practice influence teachers' beliefs and practices regarding the teaching of AI ethics. Given the challenges highlighted by teachers in this study, particularly around limited peer collaboration and lack of support, it is important to investigate how professional learning communities or teacher networks can foster shared knowledge, enhance confidence, and help teachers develop more effective teaching strategies for AI ethics. Research could explore how these communities can provide a platform for teachers to discuss best practices, share resources, and co-develop curricula, thereby potentially reshaping their beliefs about the importance and feasibility of teaching AI ethics in the classroom.

References

Ajzen, I. (1985). From Intentions to Actions: A Theory of Planned Behavior. In J. Kuhl & J. Beckmann (Eds.), *Action Control: From Cognition to Behavior* (pp. 11-39). Springer Berlin Heidelberg.

Ajzen, I. (1991). The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, *50*(2), 179-211. https://doi.org/10.1016/0749-5978(91)90020-T

Borko, H., & Putnam, R. T. (1996). Learning to teach. In *Handbook of educational psychology.* (pp. 673-708). Prentice Hall International.

Bryan, L. A. (2012). Research on Science Teacher Beliefs. In B. J. Fraser, K. Tobin, & C. J. McRobbie (Eds.), *Second International Handbook of Science Education* (pp. 477-495). Springer Netherlands.

Ham, M., & Dekkers, J. (2019). What role do teachers' beliefs play in the implementation of educational reform?: Nepali teachers' voice. *Teaching and Teacher Education*, *86*, 102917.

Handal, B., & Herrington, A. (2003). Mathematics teachers' beliefs and curriculum reform. *Mathematics Education Research Journal*, *15*(1), 59-69. https://doi.org/10.1007/BF03217369

Kagan, D. M. (1990). Ways of evaluating teacher cognition: Inferences concerning the Goldilocks Principle. *Review of educational research*, *60*(3), 419-469. https://doi.org/10.2307/1170760

Kong, S.-C., Cheung, M.-Y. W., & Tsang, O. (2024). Developing an artificial intelligence literacy framework: Evaluation of a literacy course for senior secondary students using a project-based learning approach. *Computers and Education: Artificial Intelligence*, *6*, 100214. https://doi.org/https://doi.org/10.1016/j.caeai.2024.100214

Long, D., & Magerko, B. (2020). What is AI Literacy? Competencies and Design Considerations.

Miao, F., & Shiohira, K. (2022). *K-12 AI curricula. A mapping of government-endorsed AI curricula*. https://unesdoc.unesco.org/ark:/48223/pf0000380602

Ng, D. T. K., Lee, M., Tan, R. J. Y., Hu, X., Downie, J. S., & Chu, S. K. W. (2023). A review of AI teaching and learning from 2000 to 2020. *Education and Information Technologies*, *28*(7), 8445-8501.

Pajares, M. F. (1992). Teachers' beliefs and educational research: Cleaning up a messy construct. *Review of educational research*, *62*(3), 307-332.

Stolpe, K., & Hallström, J. (2024). Artificial intelligence literacy for technology education. *Computers and Education Open*, *6*, 100159. https://doi.org/https://doi.org/10.1016/j.caeo.2024.100159

Urton, K., Wilbert, J., Krull, J., & Hennemann, T. (2023). Factors explaining teachers' intention to implement inclusive practices in the classroom: Indications based on the theory of planned behaviour. *Teaching and Teacher Education*, *132*, 104225. https://doi.org/https://doi.org/10.1016/j.tate.2023.104225

Voet, M., & De Wever, B. (2020). How do teachers prioritize instructional goals? Using the theory of planned behavior to explain goal coverage. *Teaching and Teacher Education*, *89*, 103005.

Williams, R., Kaputsos, S. P., & Breazeal, C. (2021). Teacher Perspectives on How To Train Your Robot: A Middle School AI and Ethics Curriculum. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(17), 15678-15686. https://doi.org/10.1609/aaai.v35i17.17847

Wong, G. K. W., Ma, X., Dillenbourg, P., & Huan, J. (2020). Broadening artificial intelligence education in K-12: where to start? *ACM Inroads*, *11*(1), 20–29. https://doi.org/10.1145/3381884

Wu, D., Chen, M., Chen, X., & Liu, X. (2024). Analyzing K-12 AI education: A large language model study of classroom instruction on learning theories, pedagogy, tools, and AI literacy. *Computers and Education: Artificial Intelligence*, *7*, 100295. https://doi.org/https://doi.org/10.1016/j.caeai.2024.100295

Yau, K. W., Chai, C. S., Chiu, T. K. F., Meng, H., King, I., & Yam, Y. (2023). A phenomenographic approach on teacher conceptions of teaching Artificial Intelligence (AI) in K-12 schools. *Education and Information Technologies*, *28*(1), 1041-1064. https://doi.org/10.1007/s10639-022-11161-x

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

How Teachers Enhance Young Children's Collaboration in Situated Learning Environments Through Computational Thinking Tasks

Xuechun SHI[1], Zhichun LIU[2*], Kun WANG[3]
[1]The Education University of Hong Kong, Hong Kong SAR
[2]The University of Hong Kong, Hong Kong SAR
[3]Rethink Learning Labs, United States
shixuechun2022@gmail.com, lukasliu@hku.hk, kun.wang@rethinklearning.com

Abstract: *As computational thinking (CT) becomes a fundamental literacy in the digital age, educators are increasingly focused on bringing it into real-world classrooms with young children. Based on the theorization of situated learning framing, 13 children in eastern China were involved in a two-hour activity, designing routes for a community map. This exploratory qualitative case study analyses children's behaviors and teacher intervention strategies in CT collaborative tasks through video recording and teacher interviews. The findings reveal the relationships between young children's learning (e.g., wandering, conflict, independent task division, and co-constructive problem solving) and the corresponding teachers' intervention strategies (e.g., goal reminding, resource allocation, and guiding questions). This study indicates that co-constructive problem solving is not a linear progression, but a fluctuating process marked by conflicts and setbacks. The study suggests that teachers can attempt to design tasks that require simultaneous collaboration and provide concrete demonstrations to help young children "see" the collaborative process.*

Keywords: Computational Thinking, Early Childhood Education, Collaborative Learning Task, Teacher Intervention Strategy

1. Introduction

As computational thinking (CT) becomes a fundamental literacy in the digital age, educators are increasingly focused on bringing it into real-world classrooms with young children. Sometimes, teaching CT is essential in teaching programming skills (Liu, 2023) while neglecting its nature—a systematic way of thinking to solve complex problems (Brennan & Resnick, 2012). This limitation is evident when attempting to bring CT into early childhood education: If students rely on isolated technical operations (such as writing simple commands), they may struggle to see the connection between CT and real life, let alone experience its value as a "socio-cognitive tool" (Sullivan & Bers, 2018). The integration of collaborative learning and contextualized task design offers a direction to address this gap in practice. By engaging with real-world problems and navigating their complexities, young learners can leverage technology to develop self-expression skills and strengthen their sense of community belonging (Liu et al., 2024). More importantly, they can come to understand CT as a problem-solving approach applicable to everyday life, bridging the gap between technology and real-world challenges. This study draws on a situated framing for CT (see Kafai & Proctor, 2022) as the theoretical foundation for discussing an approach to learning CT through collaboration with young children. We have designed a hybrid-media collaborative task, where children work together using art and electronic media to solve a real-world problem. In this process, especially for younger children, the role of the teacher is indispensable. In this study, we aim to explore the different intervention strategies teachers adopt in collaborative CT learning for younger children, as well as how these strategies influence students' collaborative learning. Therefore, we focus on two research questions: 1. What kinds of behaviors emerge among young children in CT collaborative tasks? 2. What are the teacher's intervention strategies for children's learning? Which strategies effectively promote children's co-constructive problem solving?

2. Theoretical Perspective

The theoretical development of collaborative learning has long centered on the question of how to achieve deep collaboration. Among these discussions, Kafai and Proctor's (2022) situated framing provides a crucial perspective for understanding the dynamics of collaboration in real classrooms. This theory emphasizes that the development of Computational Thinking (CT) must be grounded in authentic contexts, where learners naturally develop a need for collaboration by solving complex problems closely tied to real life. Existing research on collaborative CT learning in early childhood education has often been based on a cognitive framing approach, focusing on misunderstandings of programming concepts, the various challenges, and their attitudes toward programming (Zeng et al., 2023). However, this approach frequently results in independent task division: students mechanically divide tasks and ultimately piece together their work (e.g., Pugnali et al., 2017; Sullivan & Bers, 2018) without engaging in deep interactions such as goal negotiation or conflict resolution. A key limitation of cognitive framing is its focus on individual learning outcomes, overlooking social and cultural contexts (Lave & Wenger, 1991). Even in collaborative settings like pair programming, the emphasis remains on coordinating individual interactions (Campe et al., 2020). In response to this limitation, our study proposes problem-driven collaboration.

Through a situated framing, our design refines the teacher's role as a "collaboration facilitator", whose primary function is to activate children's ability to negotiate autonomously through strategic interventions (such as questioning and conflict mediation), rather than making decisions on their behalf. This non-intrusive style not only preserves young children's motivation for exploration but also combines the CT practices and the dynamics of collaboration (Wang et al., 2020) - they negotiate goals to break down tasks, refine their models through debugging to grasp "iterative optimization," and ultimately construct their understanding of CT through teamwork.

3. Research Methodology

3.1. Research Design

Based on Kafai and Proctor's theorization (2022), we designed an after-school activity in eastern China concerning the design of the navigation for a community map. The participants were 13 children aged 4 to 12, divided into four groups. The activity was observed and guided by one lead teacher and one assistant teacher. The children had one hour of group collaboration to complete their projects. During the session, the children used artistic materials (such as foam boards and light clay) along with electronic media (such as LED lights, buzzers, and small recorders) to create a 3D version of a community map. Through hands-on activities, they explored concepts such as size, shape, and orientation. Finally, the task encouraged the children to think about how to improve the navigation design to make it more user-friendly and reflect these improvements in their creations.

3.2. Data Collection and Analysis

To better understand the different behaviors in children's collaboration and the teachers' intervention strategies, we conducted an exploratory qualitative case study (Stake, 1995, 2010). The data was collected through the following methods: (1) Video recording: Capture the entire group interaction process, including verbal communication, material usage, and teacher interventions. (2) Teacher interviews: Conduct semi-structured interviews after the activity to understand the reasoning behind teachers' intervention strategies. All video and interview data were transcribed in both Mandarin Chinese and English by a researcher. To ensure the accuracy of the English transcripts and interpretations, we carried out a rigorous verification process that involved cross-checking by two researchers and participant validation to reach a consensus.

For data analysis, we used open coding and thematic analysis to construct an emerging engagement model, applying a holistic coding approach during the first cycle of coding (Saldaña, 2014). To further explore the dynamic nature of student collaboration and the corresponding teacher intervention strategies, our analysis focused on one group consisting of four students: three boys of similar age (Andy, Hugo, and David, 7-8 years old) and a younger girl (Cecilia, 4 years old). We separately coded students' learning in CT collaborative tasks (Table 1) and teacher intervention strategies (Table 2). We then mapped the teacher's intervention strategies onto different student learning and incorporated teachers' interpretations to understand their decision-making logic (Table 3). To illustrate the findings, we created a timeline visualization (Figure 1) to reveal the temporal relationship between students' learning and teacher interventions.

Table 1. Coding of student learning in CT collaborative tasks

Student Learning	Definition
Wandering	The student is unfocused, not participating in the group task, either observing passively or engaging in independent activities.
Conflict	Disagreements (e.g., conflicts for ideas) lead to verbal or behavioral opposition.
Progress Stagnation	The task is halted due to technical difficulties (e.g., broken electronic media) or cognitive limitations (e.g., spatial understanding difficulties).
Independent Task Division	Group members divide tasks without interdependency (e.g., division of labor without shared understanding and goals).
Co-Constructive Problem Solving	Group members engage in deep interactions centered around problem-solving (e.g., discussions, integrating ideas, and co-construction).

Table 2. Coding of teacher intervention strategies

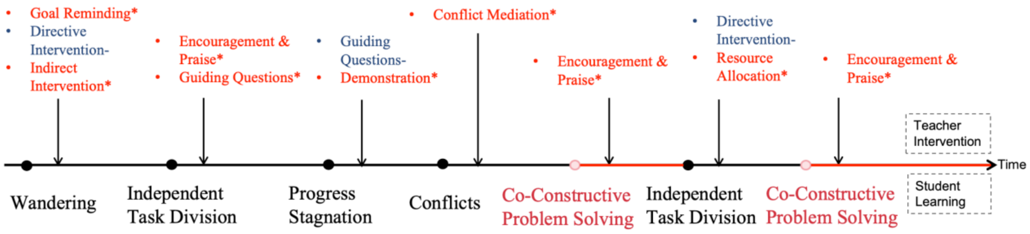
Intervention Strategy	Definition
Goal Reminding	The teacher reminds students of the task objectives through questions or prompts to refocus them on a shared problem.
Directive Intervention	The teacher gives explicit instructions for students to follow.
Indirect Intervention	The teacher encourages collaboration indirectly, such as encouraging collaboration in a way of thinking about what 'my team is doing' rather than 'what I am doing'.
Conflict Mediation	The teacher facilitates resolution by listening to both sides and helping them reach a mutual agreement.
Resource Allocation	The teacher dynamically distributes materials based on group needs to promote collaboration.
Guiding Questions	The teacher asks open-ended questions to promote students' thinking and problem-solving.
Demonstration	The teacher provides a hands-on demonstration instead of verbal instructions.
Encouragement & Praise	The teacher gives positive feedback on students' collaborative behaviors or outcomes to reinforce teamwork.

Note: the examples of categories in Tables 1 & 2 will show in Table 3

Table 3. Mapping student learning with teacher interventions

Student Learning	Teacher Intervention Strategy	Example (student learning, teachers' strategies, and results after intervention)	Interpretation from teachers (the reasons for different strategies and teachers' reflection)
Wandering	Directive Intervention -	When the teacher noticed Cecilia wandering outside the group, she gave direct instructions to Andy to assign Cecilia a task. However, Other teammates refused.	Teacher: They always felt that Cecilia wasn't capable enough—that she couldn't do this or that—so they just did everything themselves.
	Indirect Facilitation*	When the teacher saw that Cecilia was still wandering and the directive intervention was ineffective, she suggested that Cecilia observe and follow along. The group accepted this arrangement. Later, as a result, Cecilia became involved.	Teacher: I told them I wanted Cecilia to just follow along with group mates without needing to do anything. They agreed. Later, I saw that they collected a lot of boards together. That's when I knew they had started collaborating and discussing their construction plan.
Conflict	Directive Intervention -	Andy wanted to complete the entire group task by himself, but others wanted to participate, and conflict arose. The teacher tried to resolve the issue by directly assigning tasks to different students. However, Andy, as the group leader, refused to accept this approach.	Teacher: Andy is highly capable. He felt that his teammates were not helpful, so he decided he could finish everything by himself and didn't need them. So, I intervened. I completely understood Andy's perspective. I think his actions were subconscious. After receiving some guidance, he gradually started collaborating with others again, though sometimes he still struggled with it. I believe that conflicts among children are valuable. It shows us that they are still struggling with collaboration, which is a problem we need to address.
	Resource Allocation*	The teacher realized that directive intervention was ineffective in resolving this issue. Instead of directly assigning tasks, the teacher guided the students to review their current tasks and available materials, encouraging them to discuss how to distribute the materials rather than simply handing out assignments.	

Note. Due to the word limit, please refer to the website (https://osf.io/6j2qx/) for the full version. The intervention strategies marked with * were found to effectively promote collaboration, while strategies marked with - were ineffective in certain contexts.



Note: Intervention strategies marked with * were found to effectively promote collaboration, while strategies marked with - were ineffective in certain contexts.

Figure 1. Temporal relationship between students' behaviors and teachers' interventions.

4. Research Findings

4.1. The Fluctuating Pattern of Students' Learning

Young children's behaviors in these tasks follow a dynamic, fluctuating pattern rather than a straightforward progression. The study found that their behaviors progressed through "wandering → independent task division → progress stagnation → conflicts → co-constructive problem solving → independent task division → co-constructive problem solving" (conclusion by Table 3 and Figure 1).

For example, when children first encounter electronic media, they often engage in individual exploration due to the novelty of the materials. After the teacher assigns a task, they may exhibit Independent Task Division (e.g., division of labor and individual work). When they face progress stagnation (e.g., operational challenges), conflicts often become a turning point. Only through teacher guidance do they ultimately achieve co-constructive problem solving.

It is worth noting that as the activity progresses, the duration of co-constructive problem solving gradually increases (see Figure 1). However, occasional returns to independent task division may still occur. When there is a shortage of electronic media or when they are damaged, the collaborative process can be abruptly interrupted. In such cases, the children are unable to continue with their original plan, nor do they have backup materials to explore new ideas. This often leads to frustration, giving them the feeling that "this activity is simply impossible to complete (Hugo, 20min41s, personal communication)" which in turn causes them to revert to working individually in independent task division, assigning isolated responsibilities without shared accountability (Graesser et al., 2018).

4.2. The Differentiated Impact of Teacher Intervention Strategies

When students are engaged in independent task division, teachers can effectively guide them toward meaningful cooperation by using guiding questions combined with specific encouragement. This approach works because, at this stage, students still have a basic willingness to interact but lack effective collaboration methods. Providing explicit feedback on cooperative behaviors helps them develop a further understanding of why "helping each other is beneficial." When students progress to co-constructive problem solving, teachers can extend the duration of their cooperation by consistently providing process-oriented praise, which can concrete affirmation reinforces students' sense of achievement in collaboration, making them more inclined to maintain this efficient working mode.

It is important to note that directive intervention is ineffective when students are either wandering or in conflict. Such directives lead to fragmented contributions rather than a coherent solution and fall into routine coordination rather than problem-oriented interaction (Graesser et al., 2018). In contrast, when students are disengaged from the task, such as continuously handling materials without attempting to build anything, teachers can re-engage them by using goal reminding or by encouraging peer invitations. These strategies help bring the disengaged student back into the task. When conflicts arise, teachers can first listen to the reasons behind the conflict and adjust task distribution, which not only resolves opposition but also transforms the conflict into a collaborative opportunity. Observations show that groups that have undergone mediation tend to collaborate more effectively afterward.

A particularly noteworthy finding is the unexpected effectiveness of demonstration-based intervention. In the architectural modeling task, when students lacked spatial reasoning and struggled to understand "three-dimensional structures", verbal guidance alone only deepened their confusion. In such cases, a direct demonstration of 3D assembly techniques (e.g., physically attaching two foam boards perpendicularly) immediately worked as it visually presented a nice example of solutions. After witnessing the demonstration, students were not only able to replicate the basic model but also proactively added creative elements. This highlights a crucial principle: When there is a significant gap between students' cognitive level and task requirements, intuitive behavioral demonstrations are more effective than abstract verbal instructions, much like teaching a child to tie their shoes, where a hands-on demonstration is far more effective than simply describing the steps.

5. Discussion and Implication

5.1. The "Imperfect" Process of co-constructive problem solving and the Teacher's Role

This study reveals that co-constructive problem solving is not a linear progression, but a fluctuating process marked by conflicts and setbacks. Contrary to the common belief that "collaboration levels steadily improve with teacher intervention," observations show that genuine collaboration often involves disputes or even regressions. For example, in a route-design task, a group that was previously collaborating effectively might suddenly revert to working individually due to a broken wire in an electronic media or a disagreement over design choices. This fluctuating nature of collaboration highlights an important insight: Co-constructive problem solving is not a fixed skill that, once learned, is permanently mastered, which is a dynamic process that requires continuous practice. This challenges traditional expectations of the teacher's role. Instead of controlling the collaboration process from start to finish, teachers should focus on providing targeted guidance at critical moments, such as when a group reaches an impasse during a conflict. For instance, when children argue over material distribution, rather than directly assigning roles, a teacher might ask, "How can we divide the tasks so that everyone gets to participate?" This approach not only helps resolve conflicts but also nurtures students' ability to negotiate and collaborate autonomously.

To change the non-sophisticated perception of collaboration, we suggest that teachers need to work on a dual approach. On one hand, they should provide concrete demonstrations to help children "see" the cooperative process. On the other hand, they should design tasks that require simultaneous collaboration (such as a setup where two children must press different switches at the same time to light up a bulb). Through this kind of "experiential learning," children will gradually come to understand that true cooperation is not just about piecing together individual contributions, but about engaging in meaningful exchanges of ideas to discover better solutions.

References

Bers, M. U., González-González, C., & Armas–Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers and Education, 138*, 130–145. https://doi.org/10.1016/j.compedu.2019.04.013

Brennan, K., & Resnick, M. (2012). *In New frameworks for studying and assessing the development of computational thinking [Paper presentation] Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.*

Campe, S., Denner, J., Green, E., & Torres, D. (2020). Pair programming in middle school: Variations in interactions and behaviors. *Computer Science Education, 30*(1), 22–46. https://doi.org/10.1080/08993408.2019.1648119

Graesser, A. C., Fiore, S. M., Greiff, S., Andrews-Todd, J., Foltz, P. W., & Hesse, F. W. (2018). Advancing the science of collaborative problem solving. *Psychological science in the public interest, 19*(2), 59-92.

Kafai, Y. B., & Proctor, C. (2022). A revaluation of computational thinking in K–12 education: Moving toward computational literacies. *Educational Researcher*, *51*(2), 146-151. https://doi.org/10.3102/0013189X211057904

Lave, J., & Wenger, E. (1991). Situated learning: Legitimate peripheral participation. Cambridge University Press.

Liu, Z., Shi, X., & Wang, J. (2025). Fostering Young Computational Thinkers: Designing Authentic Computational Thinking Learning in Early Childhood Education Through Co-Design. *In Empowering Early Education With Computational Thinking, AI, and STEM* (pp. 309-328). IGI Global.

Liu, Z. (2023). Integrating computational thinking into K-12 education: Bridging the gap between theories and practices. *STEM Education Review*. http://dx.doi.org/10.54844/stemer.2023.0467.

Pugnali, A., Sullivan, A., & Bers, M. U. (2017). The impact of user interface on young children's computational thinking. *Journal of Information Technology Education: Innovations in Practice*, 16, 171–193. https://doi.org/10.28945/3768.

Saldaña, J. (2014). *Thinking qualitatively: Methods of mind.* Sage.

Stake, R. E. (1995). *Case study research*. Sage.

Stake, R. E. (2010). *Qualitative research: Studying how things work*. Guilford Press.

Sullivan, A., & Bers, M. U. (2018). Dancing robots: Integrating art, music, and robotics in Singapore's early childhood centers. *International Journal of Technology and Design Education, 28*(2), 325–346. https://doi.org/10.1007/s10798-017-9397-0.

Wang, X. C., Choi, Y., Benson, K., Eggleston, C., & Weber, D. (2020). Teacher's role in fostering preschoolers' computational thinking: An exploratory case study. *Early Education & Development, 32*(1), 26–48. https://doi.org/10.1080/10409289.2020.1759012.

Zeng, Y., Yang, W., & Bautista, A. (2023). Computational thinking in early childhood education: Reviewing the literature and redeveloping the three-dimensional framework. *Educational Research Review, 39*, 100520.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Reducing Summer Learning Loss in Low-Achievement Elementary Students: The Role of Learning Frequency, Continuity, and Strategy

Yu-Jhong CHEN[1]; Tzu-Chi YANG[2*]; Jiun-Yu WU[3*]; Shu-Chuan SHIH[4]
[1,2,3]National Yang Ming Chiao Tung University, Hsinchu, Taiwan
[4]National Taichung University of Education, Taichung, Taiwan
yu1026.hs12@nycu.edu.tw, tcyang_@nycu.edu.tw, jiunyu.rms@gmail.com, ssc@mail.ntcu.edu.tw

Abstract: *This study investigates the effects of summer learning behaviors on national assessment outcomes for 95,014 K1–K9 students with low learning achievements using an adaptive learning platform (TALP). Focusing on Mathematics, we investigate learners' behavior logs concerning their learning frequency, continuity, and cyclic learning patterns. The analysis of data from the assessment performance and platform usage reveals that merely increasing learning frequency does not significantly improve learning outcomes. However, continuous use of TALP content and effective cyclic learning patterns, such as balancing video viewing with exercises, are beneficial for enhancing assessment passing rates and improvement rates. These findings underscore the importance of persistent TALP learning in mitigating LLA learners' "summer learning loss" and highlight the necessity of providing regulatory instruction and learning scaffolding for low-achievement students to adaptively learn and engage with a learning platform during their summer.*

Keywords: mathematics learning, summer learning loss, self-regulated learning, TALP, learning patterns

1. Introduction

Elementary and middle school mathematics forms a crucial foundation for student learning across a range of disciplines, equipping them with essential operations, computation skills, logical reasoning, and problem-solving capabilities. However, students with low learning achievements (LLA) find mathematics particularly challenging. Notably, these students are susceptible to "summer learning loss"—a phenomenon where students experience a decline in knowledge and skills during extended school breaks. Research by Cooper et al. (2003) and Alexander et al. (2007) has shown that mathematics is especially vulnerable to this effect, exacerbating the learning disparities between LLA students and their peers as they commence a new academic year. Consequently, it is imperative to support these students in maintaining continuity in their mathematics learning over the summer to minimize knowledge loss. Adaptive learning platforms emerge as a promising tool in this context.

To identify LLA students, Taiwan's Ministry of Education (MOE) developed the Project for Implementation of Remedial Instruction - Technology-Based Testing (Priori-tbt) (MOE, 2024), a standardized assessment conducted each May and December to evaluate students' mastery of grade-level basic academic content. Students who do not meet passing standards are classified as LLA students, enabling teachers to further assist students based on assessment reports and provide timely remedial resources during the semester. Concurrently, Taiwan's MOE also developed TALP, an adaptive learning platform aligned with the national curriculum to extend learning support beyond the classroom. TALP promotes seamless learning, suitable for both in-class supplementary teaching and autonomous learning outside of school, during summer and winter vacations. The platform's usage records (logs) can align with Priori-tbt's accounts, offering empirical data on students' learning behaviors and performance, allowing us to evaluate TALP's efficacy in mitigating summer learning loss and understanding how students utilize the platform for learning.

In this study, we investigate the association between students' Priori-tbt outcomes in December with their TALP learning frequency, continuity, and effective learning patterns during the summer vacation. Therefore, this study focuses on how these factors influence low-achievement students' learning outcomes during summer learning with TALP. Our research questions are as follows: (1) Does TALP learning frequency significantly impact students' passing and improving rates? (2) Does continuity in summer learning promote improved learning outcomes? (3) Is learning pattern associated with learning outcomes? By doing so, this study might elucidate TALP's effectiveness in reducing summer learning loss for LLA students and provide insights into how we can better support these students with their learning platform engagements.

2. Literature review

2.1. Impact of Summer Learning Loss on Mathematics

Summer learning loss occurs when students experience knowledge decay due to learning breaks, with mathematics being particularly susceptible. This loss disproportionately affects LLA students, increasing the academic gap between school years (Alexander, Entwisle, & Olson, 2007; Paechter et al., 2015). Providing summer learning support can help reduce summer learning loss and maintain academic stability, making it a significant focus for supporting LLA students (Klein et al., 2024).

2.2. Summer Learning Loss and Adaptive Learning Platforms

Providing essential support, including digital resources and learning platforms, is considered an effective solution for summer learning loss (Lynch & Kim, 2017). TALP, developed by Taiwan's MOE, aligns with the national curriculum and serves as a tool for adaptive self-regulation learning outside the classroom. In addition to providing learning content, TALP offers personalized support based on student learning data (Lu et al., 2024). Unlike MOOCs, TALP is closely integrated with classroom learning, and its usage logs can be matched with the national assessment accounts, offering data on actual learning behaviors and academic outcomes. Identifying learning patterns associated with reducing summer loss can also help guide students for more effective learning. Frequency is an essential factor, while learning continuity helps students maintain learning momentum. Effective learning patterns, especially for LLA students, enhance knowledge consolidation and comprehension (Al-Bahrani, Apostolova-Mihaylova, & Marshall, 2022; Day, 2015; Malmberg, Järvenoja, & Järvelä, 2013). Accordingly, this study analyzes the learning behaviors of low-achievement students using TALP during summer to provide empirical evidence and recommendations for reducing summer learning loss.

3. Methodology

3.1. Dataset and data pre-processing

The dataset consists of 95,014 LLA students who failed the Priori-tbt in May 2023. These students' TALP usage logs from June 15 to August 31, 2023, were analyzed to examine learning behavior over the summer. After removing incomplete and extreme records, the dataset included 78,148 video viewing events, 58,505 video learning hours, and 56,811 exercise records. We measured learning frequency, continuity, and learning patterns to assess learning outcomes. Key metrics included:

- Learning Frequency: Number of video views, total video watching time (minutes), and exercise attempts.
- Learning Achievement: Passing rate (students who failed Priori-tbt in May but passed in December) and Improvement rate (students who showed improved scores from May to December).
- Learning Continuity: Maximum weeks of continuous use during summer vacation.

- Learning Pattern: Exercise/Viewing Rate (EV rate), reflecting students' preference for practice over video learning.

4. Results and discussion

4.1. Learning Frequency

The binomial logistic regression analysis showed that video views, total video viewing time, and exercise attempts did not significantly impact students' passing or improvement rates ($p > .05$), with low explanatory power (Pseudo $R^2 < .01$). This indicates that frequency alone does not directly explain variations in learning outcomes.

4.2. Learning Continuity

Our results (Figure 1) shows that Continuity in learning, which is defined as MWC (Maximum Weeks of Continuous Use), correlated with better learning outcomes. Students who continuously learned with TALP during the summer showed higher improvement rates than those who did not, with longer continuous learning associated with increased improvement rates. This suggests that learning continuity, rather than mere frequency, is a critical factor in enhancing learning achievement.

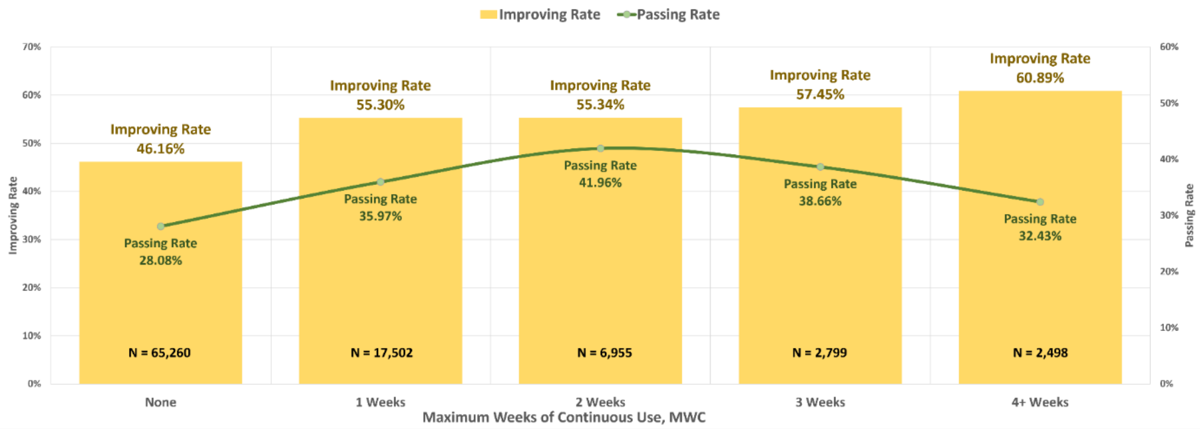


Figure 1. Comparison of Passing and Improvement Rates Among Students with Varying Levels of MWC (Maximum Weeks of Continuous Use)

4.3. Learning pattern

Analyzing changes in students' learning patterns over the summer provided insights into enhancing learning outcomes. Using the EV rate (Exercise/Viewing rate) from the two weeks before summer vacation as a baseline, we measured weekly changes in EV rate and learning frequency throughout the summer. An increased EV rate indicates an increased focus on exercise, while a decreased EV rate reflects a shift towards more video viewing. Figure 2 shows that students who failed the Priori-tbt in May but passed in December (Recovered students) increased their learning frequency for about two-thirds of the summer. These students initially emphasized video watching, later shifting to exercise attempts, suggesting a balanced learning pattern. Conversely, students who did not pass in December (LLA students) maintained a focus on video watching, with only a brief increase in practice, highlighting that learning patterns—not just frequency—may be key to progress. Similarly, Figure 3 shows that students with improved scores adopted a "video viewing-exercise" cycle, especially on weekends, underscoring the importance of regulated learning patterns over summer to boost learning outcomes.

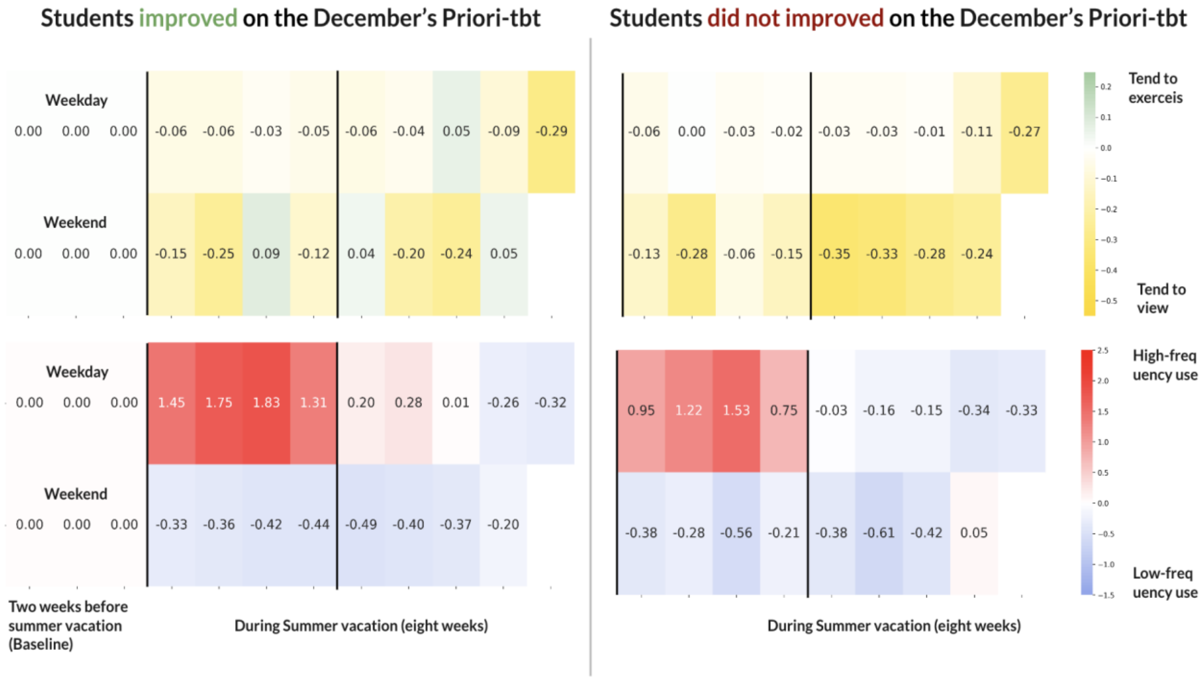


Figure 2. Comparison of Learning Patterns Between Students Who Improved and Those Who Did Not in December's Priori-tbt Assessment

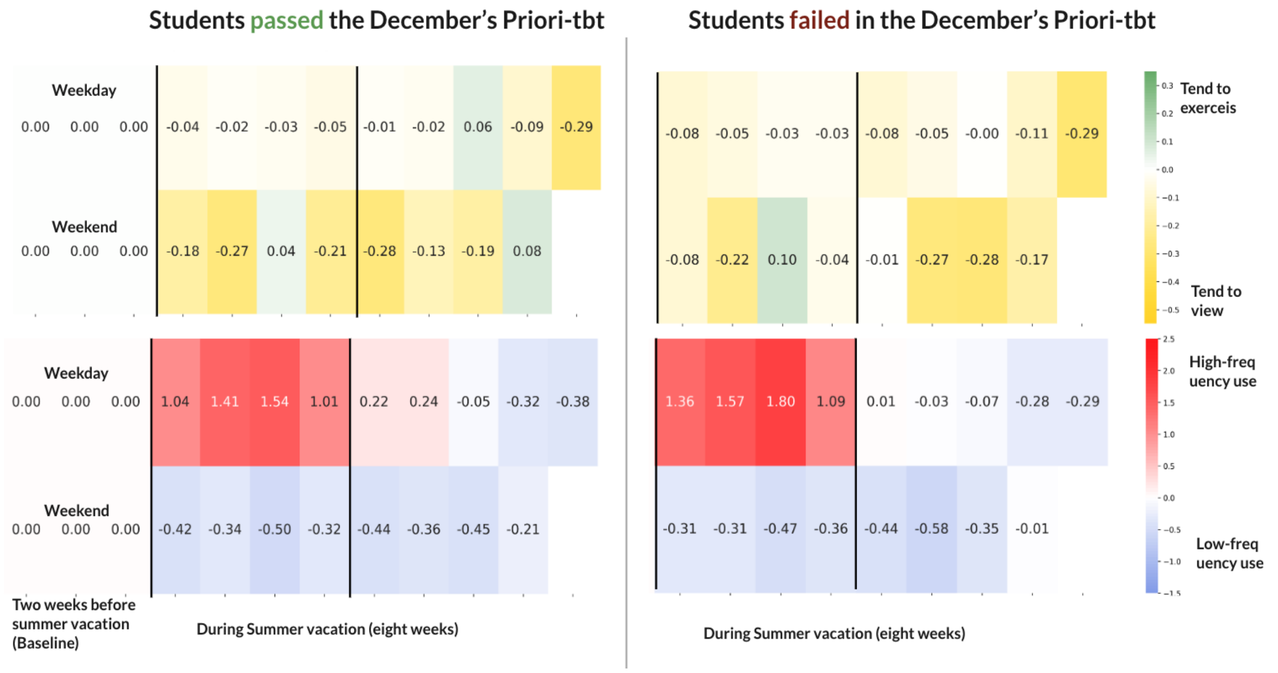


Figure 3. Comparison of Learning Patterns Between Students Who Passed and Failed December's Priori-tbt Assessment

5. Conclusion and limitations

In conclusion, this study examined the impact of summer learning through the Taiwan Adaptive Learning Platform (TALP) on the academic outcomes of LLA students, focusing on passing and improvement rates. We analyzed variables such as learning frequency, continuity, and learning patterns, leveraging comprehensive, national-level learning platform usage logs and assessment data to provide robust empirical evidence. Findings indicate that learning frequent alone does not significantly enhance learning outcomes; rather, continuous learning engagement and effective learning patterns are critical for enhance learning. Specifically, a cyclical pattern of initially increasing video views followed by exercise positively influences learning outcomes. These results highlight the value of TALP on reducing summer learning loss, particularly in supporting summer learning continuity and guiding effective learning strategies for low-achievement students.

The findings offer practical implications, suggesting that summer learning programs for LLA students should incorporate structured guidance on learning strategies and prioritize continuity in engagement. Considering the dataset scope limitations of this study, future research could investigate the learning behaviors and outcomes in different subjects and for students across various demographics, such as gender. Additionally, exploring the interaction of diverse learning strategies within adaptive platforms could further enhance the design of personalized learning supports and provide more comprehensive data insights.

Acknowledgements

This study was supported by the National Science and Technology Council in Taiwan under Grant 112-2628-H-A49-001-MY2 and 111-2410-H-A49-066-MY3. Additionally, this study was supported by the Ministry of Education, Taiwan, under the "Digital Learning Plan for Primary and Secondary Education." We sincerely appreciate their support and guidance.

References

Al-Bahrani, A., Apostolova-Mihaylova, M., & Marshall, E. C. (2022). Helping some and harming others: Homework frequency and tradeoffs in student performance. *The Journal of Economic Education*, *53*(3), 197–209. https://doi.org/10.1080/00220485.2022.2075506

Alexander, K. L., Entwisle, D. R., & Olson, L. S. (2007). Lasting consequences of the summer learning gap. *American Sociological Review*, *72*(2), 167-180. https://doi.org/10.1177/000312240707200202

Cooper, H. (2003). *Summer Learning Loss: The Problem and Some Solutions*. ERIC Digest.

Day, T. (2015). Academic Continuity: Staying True to Teaching Values and Objectives in the Face of Course Interruptions. *Teaching and Learning Inquiry: The ISSOTL Journal*, *3*, 75 - 89. https://doi.org/10.20343/TEACHLEARNINQU.3.1.75.

Gierczyk, M., & Hornby, G. (2023). Summer learning loss: review of research and implications for remediation of post-pandemic learning loss. *Preventing School Failure: Alternative Education for Children and Youth*, *67*, 132 - 140. https://doi.org/10.1080/1045988X.2023.2204823.

Klein, P. D., Casola, M., Dombroski, J., Shaw, K. W. Y., & Thompson, S. (2024). Online Intervention to Prevent Summer Learning Loss For Struggling First Grade Writers. *Reading & Writing Quarterly*, 1–25. https://doi.org/10.1080/10573569.2024.2389795

Lu, L. W., Lin, H. C. K., & Liu, H. W. (2024). Effects of the Taiwan Adaptive Learning Platform (TALP) on Learning Effectiveness and Emotion of Elementary Students with Different Cognitive Styles. *Journal of Multimedia Art Design and Education*, *4*(1), 36-50.

Lynch, K., & Kim, J. (2017). Effects of a Summer Mathematics Intervention for Low-Income Children. *Educational Evaluation and Policy Analysis*, *39*, 31 - 53. https://doi.org/10.3102/0162373716662339.

Malmberg, J., Järvenoja, H., & Järvelä, S. (2013). Patterns in elementary school students ´ strategic actions in varying learning situations. *Instructional Science, 41*, 933-954. https://doi.org/10.1007/S11251-012-9262-1.

Ministry of Education (MOE). (2024). PRIORI-tbt assessment program for elementary and junior high school learning assistance. Retrieved November 1, 2024, from https://exam.tcte.edu.tw/tbt_html/

Paechter, M., Luttenberger, S., Macher, D., Berding, F., Papousek, I., Weiss, E., & Fink, A. (2015). The Effects of Nine-Week Summer Vacation: Losses in Mathematics and Gains in Reading. *Eurasia journal of mathematics, science and technology education*, *11*, 1339-1413. https://doi.org/10.12973/EURASIA.2015.1397A.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Research on Pre-Service Training Strategies for STEM Teachers' ICT Competency

Yingqian ZHANG[1], Jiabin ZHU[2,*]
[1,2]Shanghai Jiao Tong University, Shanghai, China
zhangyingqian@sjtu.edu.cn, jiabinzhu@sjtu.edu.cn

Abstract: *The pre-service training phase directly impacts the level of ICT competency of early-career STEM teachers. Therefore, it is essential to clarify the training strategies and explore how these strategies affect early-career STEM teachers' ICT competency. This study collected data through an online questionnaire regarding the ICT competency of early-career STEM teachers and the implementation of pre-service training strategies. The research found that early-career STEM teachers' ICT competency is correlated with the six strategies at the micro level of the SQD model, which includes role model, reflection, instructional design, collaboration, authentic experience, and feedback. However, only authentic experience and feedback had a significant impact on early-career STEM teachers' ICT competency. This research provides a basis for universities to implement training strategies for enhancing STEM pre-service teachers' ICT competency.*

Keywords: pre-service, STEM teachers, ICT competency, training strategies

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

STEM 教师信息化教学能力的职前培养策略研究

章颖倩 [1]，朱佳斌 [2,*]
[1,2] 上海交通大学，上海，中国
zhangyingqian@sjtu.edu.cn, jiabinzhu@sjtu.edu.cn

【摘要】职前阶段的培养会直接表现为职早期 STEM 教师的信息化教学能力水平，因此有必要对该阶段 STEM 教师信息化教学能力的培养策略进行梳理，并探究其如何影响职早期 STEM 教师的信息化教学能力。本研究通过线上问卷的形式，对职早期 STEM 教师的信息化教学能力及职前阶段培养策略的实施情况进行数据收集。研究发现，职早期 STEM 教师的信息化教学能力与榜样作用、合作、反思、教学设计、实习实践和反馈这六个策略均存在相关性，但只有实习实践和反馈会对其信息化教学能力有显著作用。研究为高校实施 STEM 教师信息化教学能力的培养策略提供了一定的依据。

【关键词】职前，STEM 教师，信息化教学能力，培养策略

1. 引言

信息技术既是 STEM 教育的重要教学内容，又是 STEM 教育的重要教学手段。STEM 教师的信息化教学能力能够有效促进信息技术、STEM 教学和学科内容知识的有机整合，因此培养 STEM 教师的信息化教学能力具有重要意义。高校作为教师培养的重要场所，被期望能够在职前阶段有效培养 STEM 教师的信息化教学能力。然而当前 STEM 教师的培养仍以分科式的学科教师培养为主，现有信息化教学能力培养策略是否符合 STEM 教学的特色仍然存疑。此外，师范院校 STEM 教师培养项目在开展中常设计有教育技术相关的课程，但一般综合性院校则欠缺相关的经验。职前阶段的培养往往直接表现为教师在职早期阶段的信息化教学能力。研究者们尝试设计 STEM 教师信息化教学能力的培养项目并提出针对性的培养策略，但现有理论和模型能否很好地解释并总结针对 STEM 教师信息化教学能力的培养策略，仍缺少直接的证据支持。因此，本研究以中国浙江省杭州市初中阶段职早期 STEM 教师为研究对象，旨在调查和分析职前阶段的培养策略对初中阶段职早期 STEM 教师信息化教学能力的影响，以便为高校开展对 STEM 职前教师信息化教学能力的培养提供依据。

2. 文献综述

2.1. 教师的信息化教学能力

因信息技术对教育的深刻影响，不同机构和组织均发布了信息化教学能力的标准，来保证教师信息化教学能力的有效培养。如联合国教科文组织制定的 ICT-CFT 框架，认为教师的信息化教学能力应该包含信息技术相关的知识和技能以及利用信息技术进行教学实践的能力。欧盟的 DigcompEdu 框架将教师的信息化教学能力定义为熟练掌握信息技术相关的知识和技能并能够较好地使用信息技术进行教学的能力。中国教育部发布的《中小学教师信息技术应用能力标准（试行）》认为教师的信息化教学能力指的是教师应用信息技术优化课堂教学及转变学习方式的能力。

理论层面，已有学者对教师信息化教学能力的内涵、模型开展了大量的研究。其中，TPACK 框架常被认为可以为教师信息化教学能力的测量提供理论依据（Ferrari，2012）。TPACK 框架认为教师的信息化教学能力指的是教师将技术知识（TK）整合进学科内容知识（CK）和教学法知识（PK），并进行综合教学的能力（Mishra & Koehler, 2006）。TPACK 框架综合考虑了教

师进行信息化教学时的所有要素，能够对其能力进行系统全面的分析。因此有较多学者根据 TPACK 框架展开对教师信息化教学能力的探索和评估，并设计了一系列包括标准化的自我评分量表、开放式问卷、访谈和绩效评估等的评价方法（Gur, 2016）。基于此，已经有学者尝试结合 TPACK 框架和 STEM 教育并设计了 TPACK-STEM 量表，用以调查中国 STEM 教师的信息化教学能力（Chai et al, 2020）。

2.2. 信息化教学能力培养策略模型

信息技术在 STEM 教育中的发展，要求 STEM 教师必须具备一定水平的信息化教学能力。合理且有效的培养策略的实施，能够在职前阶段有效提升 STEM 教师的信息化教学能力，并促进其可持续发展。早在 2012 年，Tondeur 等人（2012）通过对当时已有的 19 项关于教师信息化教学能力培养策略的研究进行归纳性总结，对多项关于职前教师信息化教学能力培养策略的研究的分析和结论进行汇总，提出了职前教师信息化教学的 SQD（Synthesis of Qualitative Data）模型。该模型将培养教师信息化教学能力的培养策略分为宏观宏观的制度层面、中观的组织层面和微观的实施策略层面这三个层面，其中微观层面的六个策略（榜样作用、合作、反思、教学设计、实习实践、反馈）往往对 STEM 教师信息化教学能力的培养产生直接的影响，且能够设计相应的题项进行测量（Tondeur et al., 2012）。

有学者将 SQD 模型微观层面的量表引入 STEM 教育，来对 STEM 职前教师信息化教学能力培养策略实施情况进行调查。如有学者发现，树立榜样是教师信息化教学能力发展的重要动力，同行对信息技术使用，会直接影响教师信息化教学的意向（Israel et al.，2015）。然而，只考虑单一策略并不足以培养教师的信息化教学能力，要培养 STEM 教师的信息化教学能力还需要不断反思信息技术在 STEM 教育中的使用，同时辅之以相关的技术课程，并在 STEM 教学实践中对信息技术进行使用等等。单一策略的良好体现并不足以对 STEM 教师的信息化教学能力产生全面影响，微观层面的策略更多是相辅相成、相互促进的关系（Tondeur et al., 2016）。因此，要培养 STEM 教师信息化教学能力，有必要厘清这六个策略之间的关系，并进行综合考虑（Tondeur et al., 2021）。

2.3. 研究假设及理论框架

上分分析可知，合理的培养策略的实施，可以有效促进 STEM 职前教师的信息化教学能力。因此，本研究主要基于 Tondeur 等人（2012）提出 SQD 模型，探究其微观层面的六个策略对 STEM 教师信息化教学能力的影响，并提出如下假设：

H1：榜样作用能够促进 STEM 教师的信息化能力。

H2：合作能够促进 STEM 教师的信息化能力。

H3：反思能够促进 STEM 教师的信息化能力。

H4：教学设计能够促进 STEM 教师的信息化能力。

H5：实习实践能够促进 STEM 教师的信息化能力。

H6：反馈能够促进 STEM 教师的信息化能力。

根据 SQD 模型提出上述假设，分析模型如下图 1 所示。

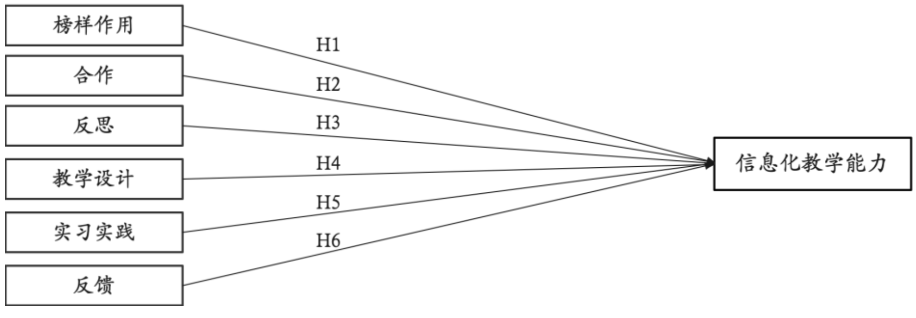


图 1 分析模型

3. 研究方法

本研究旨在探究职前阶段的培养策略对职早期 STEM 教师信息化教学能力的作用，因此研究采用定量研究的方法，通过线上问卷（问卷星）的形式对量化数据进行收集。

3.1. 研究对象

本研究选取浙江省杭州市初中阶段职早期 STEM 教师作为研究对象。通过杭州市西湖区教研院，以线上问卷（问卷星）的形式向辖区内初中学校的科学、数学教师、信息技术和工程类教师发放问卷。根据杭州市教委规定，教龄在 5 年内的教师统称为"新教师"，处于其教育生涯的早期阶段，需要接受各区教委的新教师培训。同时，根据区政务网公布数据，西湖区初中阶段符合要求的 STEM 教师人数约在 700-800 左右。本次研究共计回收问卷 231 份，其中有效问卷 227 分，占问卷总数的 98%。其中数学教师 68 人，科学教师 116 人，技术类教师 36 人，工程类教师 7 人。人数及比例分布符合当前学校教师的教师组成。

3.2. 问卷设计

问卷主要分为三个部分。第一部分主要依据 Schmid 等（2020）设计的基于 TPACK 框架的教师信息化教学能力量表。第二部分主要采用 Tondeur 等（2016）开发的 SQD 量表，从榜样作用、合作、反思、教学设计、实习实践以及反馈这六个方面对职早期 STEM 教师教师信息化教学能力在职前阶段的培养策略实施情况进行数据收集。第三部分为个人信息部分，包括受访者的性别、年龄、学历、毕业院校、专业、教授科目、教授年级以及教龄等基本信息。问卷采用李克特五分量表的形式，"完全不同意"至"完全同意"分别用"1"到"5"表示。个人信息部分，男性为"1"，女性为"2"。年龄根据题项设计由低到高分别用"1"至"4"表示。同样的学历、教授科目、教授年级和教龄均根据题项设计用数字进行表示。

3.3. 数据分析

本研究使用 SPSS 软件对量化数据进行分析。为表明本研究设计量表在中国 STEM 教育情境下的适用性，研究人员对量表的信效度进行分析。随后，在问卷信效度分析的基础上，研究人员运用回归分析探索职早期 STEM 教师职前阶段的信息化教学能力培养策略与教学能力之间的因果关系。

4. 研究结果

4.1. 信效度分析

研究人员根据问卷所得数据，对问卷的信效度进行分析，所得数据如下表 1 所示。表中数据表明，问卷信效度较好，可以用于研究的数据收集和进一步研究的开展。

表 1 量表信效度

分量表	克隆巴赫（α 值）	KMO
知识/技能	0.789	0.736
职前培养策略	0.869	0.905

4.2. 相关性分析

本研究以职前阶段培养在微观层面的 6 个培养策略为自变量，分别是榜样作用、教学设计、实习实践、合作、反思和反馈。职早期 STEM 教师的信息化教学能力为应变量。控制变量为职早期 STEM 教师的性别、年龄、学历、教授的科目、教授的年级以及教龄。对培养策略和职早期 STEM 教师的信息化教学能力进行相关性分析，结果整理后如下表 2 所示。表中数据显示职早期 STEM 教师的信息化教能力与榜样作用、合作、反思、教学设计、实习实践和反馈均存在相关性。同时，STEM 教师的教授年级和教龄与其信息化教学能力也相关。

表 2 相关性分析结果呈现

	性别	年龄	学历	科目	年级	教龄	信息化教学能力	榜样作用	合作	反思	教学设计	实习实践	反馈
性别	1	-0.164*	0.137*	0.260**	-0.211**	-0,224**	-0.102	0.058	0.018	-0.034	-0.006	-0.039	-0.030
		0.013	0.040	0.000	0.001	0.001	0.127	0.383	0.786	0.612	0.925	0.561	0.657
年龄		1	0.274**	-0.265**	0.276**	0.522**	0.060	-0.092	-0.055	0.110	0.037	-0.049	0.094
			0.000	0.000	0.000	0.000	0.367	0.166	0.410	0.098	0.576	0.465	0.160
学历			1	-0.123	0.080	-0.014	0.101	0.176**	0.094	0.127	0.065	0.037	0.115
				0.064	0.228	0.832	0.128	0.008	0.159	0.056	0.326	0.575	0.083
科目				1	0.289**	-0.072	-0.073	-0.045	-0.038	-0.039	-0.120	0.030	-0.061
					0.000	0.283	0.271	0.498	0.572	0.557	0.071	0.649	0.363
年级					1	0.337**	0.142*	0.099	0.031	0.091	0.089	0.131*	0.079
						0.000	0.033	0.138	0.638	0.174	0.181	0.049	0.236
教龄						1	0.229**	0.013	0.081	0.139*	0.198**	0.201**	0.213**
							0.000	0.847	0.225	0.036	0.003	0.002	0.001
信息化教学能力							1	0.470**	0.538**	0.519**	0.578**	0.670**	0.522**
								0.000	0.000	0.000	0.000	0.000	0.000
榜样作用								1	0.611**	0.520**	0.480**	0.504**	0.287**
									0.000	0.000	0.000	0.000	0.000
合作									1	0.523**	0.690**	0.569**	0.449**
										0.000	0.000	0.000	0.000
反思										1	0.601**	0.593**	0.489**
											0.000	0.000	0.000
教学设计											1	0.653**	0.645**
												0.000	0.000
实习实践												1	0.535**
													0.000
反馈													1

注：*表示 p<0.05，**表示 p<0.01，***p<0.001。

4.3. 回归分析

研究采用 SPSS 软件，对提出的假设进行检验。回归结果如表 3 所示。

表 3 回归分析结果呈现

模型	未标准化系数 B	未标准化系数 标准误差	标准化系数 Beta	t	显著性
常数项	1.832	0.187		9.817	0.000
性别	-0.038	0.029	-0.067	-1.313	0.190
年龄	0.003	0.025	0.009	0.136	0.892
学历	0.018	0.028	0.034	0.653	0.514
科目	-0.011	0.020	-0.031	-0.584	0.560
年级	0.002	0.020	0.004	0.084	0.933
教龄	0.015	0.013	0.069	1.137	0.257
榜样作用	0.073	0.045	0.104	1.610	0.109
合作	0.077	0.047	0.121	1.664	0.098
反思	0.032	0.040	0.052	0.791	0.430
教学设计	0.018	0.045	0.033	0.405	0.686
实习实践	0.276	0.049	0.397	5.628***	0.000
反馈	0.073	0.030	0.155	2.433*	0.016

从回归结构可知，实习实践对 STEM 教师信息化教学能力的影响显著（β=0.397，p<0.001）。数据可以说明实习实践与 STEM 教师的信息化教学能力呈现正相关，说明信息化教学相关的实习实践越多，STEM 教师的信息化教学能力就越高。假设 H5 得到验证。此外，反馈对 STEM 教师信息化教学能力的影响显著（β=0.155，p<0.05）。数据也能说明反馈与 STEM 教师的信息化教学能力呈现正相关，说明 STEM 教师在职前阶段收到的反馈越多，相应的信息化教学能力也越高。假设 H6 得到验证。假设 H1-H4 未能得到验证，研究结果只证明了榜样作用、合作、反思和教学设计与 STEM 教师的信息化教学能力相关，但并不存在直接的因果关系。同时与 STEM 教师信息化教学能力相关的教授年级和教龄也与其不存在直接的因果关系。

5. 讨论与启示

本研究探究发现实习实践会对 STEM 教师的信息化教学能力产生显著影响。实习实践是 STEM 教师在职前阶段信息化教学能力的重要实践环节，STEM 职前教师可以将他们的的信息化教学知识和技术应用于真实环境中(Valtonen et al., 2015)。实习实践过程中的沉浸式参与经验能够促进 STEM 职前教师更好地理解理论与教学实践之间的联系，从而促进其信息化教学能力的有效提升（Tondeur et al. 2016）。同时本研究发现 STEM 教师在职前阶段接受到的信息化教学相关的反馈会对其信息化教学能力产生促进作用。STEM 教师在整个职前培养过程中接受到的来自老师和学生的持续反馈，能够帮助 STEM 职前教师在培养过程中对自身的信息化教学能力的理解和思考，并进行持续不断的修正和提升，从而提升 STEM 职前教师在课堂教学中对信息技术的使用（Banas & York, 2014）。

Tondeur 等（2012）基于文献总结了教师信息化教学能力的职前培养策略模型，并指出微观层面的六个策略应进行整合思考。如实习实践过程中还会涉及到 STEM 课堂设计、带教老师的指导和反馈，同学之间的互相学习与合作等等。这可能导致本研究的结果只有实习实践和反馈产生了直接的作用，并不表示其他策略不对 STEM 教师的信息化教学能力产生影响。微观层面的策略在很多时候是互相嵌套无法进行明确分割的。后续可以通过增加样本量，在更大范围内进行本研究问卷的发放，并对数据进行继续分析。同时，可以尝试用结构方程模

型（Structural Equation Model，SEM）对量化数据进行进一步的路径分析，用以厘清六个策略共同作用于职早期 STEM 教师信息化教学能力的具体路径。

致谢

本研究由教育部人文社会科学研究一般项目（项目编号：22YJA880096）资助。

参考文献

Banas, J. R., & York, C. S. (2014). Authentic learning exercises as a means to influence preservice teachers' technology integration self-efficacy and intentions to integrate technology. *Australasian Journal of Educational Technology, 30*(6).

Chai, C. S., Rahmawati, Y., & Jong, M. S. Y. (2020). Indonesian science, mathematics, and engineering preservice teachers' experiences in STEM-TPACK design-based learning. *Sustainability, 12*(21), 9050.

Ferrari, A. (2012). *Digital competence in practice: An analysis of frameworks* (Vol. 10, p. 82116). Luxembourg: Publications Office of the European Union.

Gur, H. (2015). A short review of TPACK for teacher education. *Educational Research and Reviews, 10*(7), 777-789.

Israel, E., Duplàa, E., Kaufman, D., Sauvé, L., & Renaud, L. (2015, June). Investigating the link among flow, social interaction and ICT skills with WOW. In *EdMedia+ Innovate Learning* (pp. 415-425). Association for the Advancement of Computing in Education (AACE).

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers college record*, *108*(6), 1017-1054.

Schmid, M., Brianza, E., & Petko, D. Developing a short assessment instrument for Technological Pedagogical Content Knowledge (TPACK. xs) and comparing the factor structure of an integrative and a transformative model. *Computers & Education*, 2020, *157*, 103967.

Tondeur, J., Van Braak, J., Sang, G., Voogt, J., Fisser, P., & Ottenbreit-Leftwich, A. (2012). Preparing pre-service teachers to integrate technology in education: A synthesis of qualitative evidence. *Computers & education*, *59*(1), 134-144.

Tondeur, J., Van Braak, J., Siddiq, F., & Scherer, R. (2016). Time for a new approach to prepare future teachers for educational technology use: Its meaning and measurement. *Computers & Education*, *94*, 134-150.

Tondeur, J., Howard, S. K., & Yang, J. (2021). One-size does not fit all: Towards an adaptive model to develop preservice teachers' digital competencies. *Computers in Human Behavior*, *116*, 106659.

Valtonen, T., Kukkonen, J., Kontkanen, S., Sormunen, K., Dillon, P., & Sointu, E. (2015). The impact of authentic learning experiences with ICT on pre-service teachers' intentions to use ICT for teaching and learning. *Computers & Education*, *81*, 49-58.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Working to Foreground Relationality in Computational Thinking

Ezra GOUVEA[1], Ali ASIF[2], Rohini THAPA[3], Kolawole KUSHIMO[4], Chandra ORRILL[5], Ramprasad BALASUBRAMANIAN[6], Shakhnoza KAYUMOVA[7]
[1,5]Rethink Learning, Inc., United States
[2,3,4,6,7]University of Massachusetts, Dartmouth, United States
ezra.gouvea@rethinklearning.com, rthapa@umassd.edu, kkushimo@umassd.edu, aasif@umassd.edu, chandra.orrill@rethinklearning.com, r.bala@umassd.edu, skayumova@umassd.edu

Abstract: *Current discourses about the concept of computational thinking (CT) are shaping what we decide are the components of CT, shaping the ways we explain why CT is important for students, and shaping how CT will evolve as a disciplinary standard. We contribute to this conversation by investigating where CT resonates with problematic societal discourses, and how CT can be (re)understood to be more compatible with systems of relational ethics. Using a concept of pluriversal design for science learning spaces, we examine data from a teacher development program with an objective of reimagining and rearticulating the meaning of computational thinking practices. In this paper, we rearticulate the concept of "debugging" to foreground relational practices of close attention and attunement to relationships and interconnectedness. This work contributes to scholarship working to bridge the current conceptual distance between computational thinking practices and practices of ethical relationality.*

Keywords: debugging, relational ontology, postfoundational research, deficit narratives

1. Introduction & Purpose

Debugging is a key component of Computational Thinking (CT) and is generally described as a systematic approach to finding and fixing bugs in the system (Grover & Pea, 2013; Weintrop et al., 2016; Wing, 2006). Widely considered an essential problem-solving skill (Liu et al., 2017), there is considerable evidence from research (Carver & Risinger, 1987; Klahr & Carver, 1988) that debugging programs or codes can enhance general troubleshooting skills in other non-programming domains.

Yet, we notice that definitions of debugging (as well as other aspects of CT) often resonate with deficit-based societal discourses. Although perhaps subtly, a part of what it means to be a computational thinker is to be someone who straightforwardly decides what a program is supposed to do, confidently asserts what code belongs to serve their purposes, and expertly removes "bugs" that behave in ways that they decide is objectionable to their interests. We notice somewhat similarly that the dominant framing for what it means to practice abstraction in Computer Science is to become an arbiter of what "really" matters, to make authoritative decisions of what can and should be discarded, and to process knowledge into forms that can then be easily transported to other places (Lafuente Martínez et al., 2022; Rijke et al., 2018). With a narrow focus on computer programming, we recognize that it is difficult to see these framings as anything but harmless. But expanding our attention to notice broader similarities with other problematic patterns of societal discourses, we see these framings as uncomfortably resonant with the logics that perpetuate systemic oppression.

If computational thinking practices carry potential for sustaining problematic societal discourses, then teacher development programs also carry invisible potentials for developing teachers that are susceptible to these problematic logics. *In this paper, we wish to reimagine and rearticulate the concept of debugging to resist these broader deficit discourses.*

1.1. Theoretical Framework

Kayumova and Dou (2022) draw upon the concept of a pluriverse—a "world where many worlds fit" (Escobar, 2018, p. xi)—to "interrogate why different ways of being…are constantly figured in deficit terms within the current institutions of science" (Kayumova & Dou, 2022, p. 1098). They argue that "theories and theoretical frameworks manifest onto-epistemological beliefs and identities and are therefore lenses through which we can see, analyze, and build new visions and justice-oriented worlds" (p. 1113). To disrupt deficit-based discourses, they urge that "we must examine the logic upon which dominant science-related epistemological and ontological assumptions are built" (p. 1100) and they argue that "equity and justice are about *changing/transforming science structures and norms* by forging new asset-based and relational justice oriented spaces" (p. 1100, emphasis added).

1.2. Contemporary Scholarship

Justice-oriented scholarship in STEM Education has increasingly begun to emphasize the importance of expecting and respecting multiple ways of knowing (Rosebery et al., 2010; Warren & Rosebery, 2011), and research in Computer Science education has also begun to incorporate these perspectives. Sengupta, Dickes, and Farris (2018) have argued for an epistemological shift towards understanding computational thinking as phenomena composed of discursive, perspectival, material, and embodied experiences. Across multiple publications, these three authors advocate for a Bakhtinian reframing of computational thinking as a dialogical co-construction of computational utterances (Sengupta et al., 2021). Building on the perspective that "computational thinking must be re-conceptualized more appropriately as an intersubjective experience" (Sengupta et al., 2018, p. 29), Dickes and Farris argue for a shift away from conceptualizing CT as a set of isolated material intelligences and to instead focus on the "complex interplay among materials (both computational and noncomputational), cognition, and classroom culture" (Dickes & Farris, 2019, p. 8).

One aspect of this complex interplay is the entanglement of computational thinking, empathy, and ethics. Sohr, Gupta, Elby, and Radoff (2023) describe a student's entanglement of empathy and computational thinking in reasoning about fairness. Their analysis highlights the inherent ethical decisions that are entangled with the computational practices of abstraction (that determine which considerations are worth of attention). Silvis, Clarke-Midura, Shumway, and Lee (2022) advocate for incorporating an ethic of technological care into other ways of relating—becoming a central tenant of any ethical discourse of computing. Their feminist analysis of the entangled ethical and computational practices recognizes the important consequences of critically examining who is regarded as worthy of care (Haraway, 2020).

Something that is implied by analyses such as these (but not always explicitly mentioned) is that the *absence* of a central tenant of technological care within current discourses of computing simultaneously mobilizes a complementary disciplinary ethic: Maintaining a distance between discourses of computing and care inherently establishes a default ethic of ambivalence in computer science. Without ethics of care built into the disciplinary practices of computer science, ignorance of the impacts of our actions is instead left behind as the default enactment of disciplinary ethics for computing.

2. Problem & Objective

Despite these ongoing developments of contemporary research, descriptions of CT most often resonate to some extent with deficit discourses and do not straightforwardly align with a pluriversal approach. First, CT is often characterized as a universal skill set that will enrich everyone's lives. This type of statement of universality is in tension with the concept of a pluriverse. Deficit-based discourses require a constructed "universal" zero-point (Warren et al., 2020) against which to measure/compare the deficit. The purpose of creating space for heterogeneous ways of knowing (or a pluriverse of onto-epistemologies) is precisely to disrupt that mechanism because it perpetuates harm. As mentioned above, descriptions of CT practices (or STEM practices) are sometimes uncomfortably resonant with more obviously problematic logics. Such resonances are additional impediments to a pluriversal approach, since we are urged to question the underlying logic of science and forge new asset-based science structures and norms. There is no need to accept these resonances between CT and logics that sustain oppression. We notice that descriptions of debugging are resonant with settler-colonial logics (Simpson, 2017) of elimination of anything deemed a nuisance (without regard for the impact on ecosystems). We assert that an accounting of exactly how much harm is caused by these resonances is not necessary because *any potential for resonance should be considered sufficient for engaging in the work of reexamining how we tune the system.*

In this paper, our objective is to investigate whether debugging can be framed more compatibly with heterogeneity and pluriversality. Drawing from ontologies of care and relationality, Kayumova and Dou (2022) suggest that pluriversal design could include "reimagining and rearticulating meanings of science learning, being a science person, and engaging in science as people who are caring, interconnected, respectful, humanizing, honoring, reverent, just, and dignifying" (p. 1110). *With this pluriversal sensibility in mind, we examine a learning environment with the objective of reimagining and rearticulating the meaning of "debugging" as a practice of close attention and attunement (Shotter, 2015) to relationships and interconnectedness.*

3. Setting & Mode of Inquiry

Our engagement with data adheres to standards of Interaction Analysis (Jordan & Henderson, 1995) (but we also note an important divergence). Our research project involved developing and implementing a professional development program for teachers to both learn about computational thinking and develop their own lessons incorporating computational thinking practices into their curriculum. We videorecorded approximately 90 hours of professional development over two separate year-long sequences. Each year, *participating teachers attended a week-long workshop over the summer, monthly professional development sessions during the school year, and monthly one-on-one coaching sessions* with the workshop instructors with feedback on their lesson planning.

Relational approaches such as pluriversality need to be compatible with systems of ethics that recognize an entanglement of ethics (axiology), epistemology, and ontology. Although discursively framed in a plurality of ways, such relational ontologies necessarily engage with a poststructural critique of representationalism (Barad, 2003; Rosiek & Gleason, 2017). This critique leads to a shift of approaches towards research methodology (Garroutte & Westcott, 2013; Higgins & Kim, 2019; Murris, 2020). Some scholars have chosen to describe their research as *postfoundational* or postqualitative (Dixon et al., 2023; Lather & St. Pierre, 2013) due to this incompatibility of poststructuralist philosophy with structuralist qualitative methodology (which typically involves finding or constructing measures of the world and building representations that are categorized as passive or inert). Drawing from both new materialist philosophy and Indigenous studies scholarship, Rosiek and Snyder (2020) use the term *narrative inquiry* to refer to approaches that acknowledge the agency of methodology, results, analysis, theory, etc. Rosiek and Snyder describe narrative inquiry "as a process of reimagining the possibilities within experience that ontologically transforms a person's relation to his or her vocational activity" (2020, p. 1158).

Our narrative of debugging diverges from structuralist enactments of Interaction Analysis because we are positioning it as postfoundational research. *This paper performs a postfoundational narrative inquiry built to help reconstitute "the way we frame our questions and by the material features of our inquiry apparatuses" (Rosiek & Snyder, 2020, p. 1152).* Debugging is an underrepresented topic in K-12 classrooms (Kafai et al., 2020; Michaeli & Romeike, 2019; Rich et al., 2019), and there is a gap in the literature examining exactly how debugging emerges for K–12 teachers and students. One aspect of our data analysis has been to search for episodes of debugging that happened during the professional development sessions. We identified such a moment of debugging during one of the monthly professional development sessions, and noticed that it was novel in that the episode did not quite fit the standard narrative of debugging. Through repeated video review and event reconstruction, we developed transcriptions and analytical memos to describe the interactions that inhered in the teachers' and instructor's debugging process. We refined our analytical memos into a narrative that intentionally crafts a different framing for debugging (i.e., characterized as close attention to the ways many things relate and influence each other)—because the objective for our inquiry is to find discourses for debugging that are compatible with pluriversality.

4. Narrative of Debugging as Understanding Networks of Relations

Two of our teachers (Mary and Julie) and the workshop instructor (Jarod) found themselves debugging a system involving a small mobile robot programmed in a block-based coding environment. They want the robot to do something, and at first the robot was not doing it. They spent time trying to understand and adjust. And then the robot did what they had hoped. So, we can confidently say that whatever happened in their time understanding and adjusting can be characterized as "debugging." The events that happened can be captured equally well—if not better—by a narrative of the participants coming to understand the nuances of the relationships among elements of the system.

The robot has a sensor on its "chest" that uses a calibrated frequency of light to measure the distance to an object in front of it. And the group was attempting to program this robot so that it would check whether something was blocking its way, and then either continue ahead or turn around and move in the opposite direction. Julie's code was not making the robot behave how they wanted. It continued ahead without turning around when it should have. Jarod then changed the code. The robot's behavior running the code was to turn around without continuing ahead when it should. The group questioned whether having the robot on the floor was affecting the response of the sensor and causing the robot to behave how they observed. The group then put the robot on the table and ran the code while Jarod held his arms in front of the sensor. The robot continued ahead into Jarod's arms. But the same code then caused the robot to repeatedly turn around when nothing was in front of the sensor. Jarod questioned whether his previous understanding of the code syntax was correct, and the group changed the code back to Julie's original version. The robot then responded in the way they had intended to program it.

What we find interesting is that this debugging process involved the group examining the ways that all the elements of the system were coming together. We are resistant to identify any one "bug" that they needed to identify and eliminate. Julie's original code was logically correct, and her understanding of the conditional statements was correct. The sensor was not broken, it was working correctly; the materiality of the sensor just is the way that it is. Jarod's temporary understanding of the syntax of the coding environment could be considered wrong. But that error did not cause the problem; so we are resistant to calling that "the bug." Their reasonable expectations for how the sensor would react with the robot on the floor could possibly be identified as the "bug." But we are resistant to characterizing their debugging process as systematically eliminating that expectation. The data would not fit that description any better than a relational version of the debugging process. We suggest that in order to understand how the sensor, code, and surface were interacting, the group engaged in a creative process of imagining what could be happening to the agents in the system. The group used and modified what they knew about the agents in relation to understand more deeply how they interrelated. Describing debugging in this way as a process of paying close attention in order to understand the subtle ways that many things interrelate and influence each other fits this scenario just as well as other dominant narratives of debugging.

5. Conclusion & Significance

We presented an example of debugging from a professional development workshop for teachers. We argued that what happened can be characterized as an episode of coming to understand how elements of the system were interrelated. As compared to "making the world conform to what an authority figure has decided is correct," we notice that teacher development that foregrounds the relationality of debugging—as "deeply understanding relations and repairing them where necessary"—has more potential to support an ethical future.

The significance of this work relies on recognizing and problematizing the current large conceptual distance between "care" and "debugging" within dominant discourses. The purpose of this work is not to suggest that the concept of debugging has always or already included care as a feature, but to instead argue that *we have agency to choose and shape how disciplinary standards in Computer Science resonate with other societal discourses.* And we have an opportunity to iteratively rearticulate the practices of science so that we develop ourselves as teachers who continuously become better at caring for and repairing relations.

References

Barad, K. (2003). Posthumanist performativity: Toward an understanding of how matter comes to matter. *Signs: Journal of Women in Culture and Society*, *28*(3), 801–831.

Carver, M. S., & Risinger, S. C. (1987). Improving children's debugging skills. In *Empirical studies of programmers: Second workshop* (pp. 147–171). Ablex Publishing Corp.

Dickes, A. C., & Farris, A. V. (2019). Beyond Isolated Competencies: Computational Literacy in an Elementary Science Classroom. In P. Sengupta, M.-C. Shanahan, & B. Kim (Eds.), *Critical, Transdisciplinary and Embodied Approaches in STEM Education* (pp. 131–149). Springer International Publishing. https://doi.org/10.1007/978-3-030-29489-2_8

Dixon, K., Murris, K., Peers, J., Giorza, T., & Lawrence, C. (2023). *Postdigital Play and Global Education: Reconfiguring Research*. Routledge.

Escobar, A. (2018). *Designs for the Pluriverse: Radical interdependence, autonomy, and the making of worlds*. Duke University Press.

Garroutte, E. M., & Westcott, K. D. (2013). The story is a living being: Companionship with stories in Anishinaabeg studies. *Centering Anishinaabeg Studies: Understanding the World through Stories*, 61–80.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Haraway, D. J. (2020). *Staying with the Trouble: Making Kin in the Chthulucene*. Duke University Press. https://doi.org/10.1515/9780822373780

Higgins, M., & Kim, E.-J. A. (2019). De/colonizing methodologies in science education: Rebraiding research theory–practice–ethics with Indigenous theories and theorists. *Cultural Studies of Science Education*, *14*(1), 111–127.

Jordan, B., & Henderson, A. (1995). Interaction analysis: Foundations and practice. *The Journal of the Learning Sciences*, *4*(1), 39–103.

Kafai, Y., Biswas, G., Hutchins, N., Snyder, C., Brennan, K., Haduong, P., DesPortes, K., Fong, M., Flood, V. J., Aalst, O. W., DeLiema, D., Fields, D., Gresalfi, M., Brady, C., Steinberg, S., Knowe, M., Franklin, D., Coenraad, M., Weintrop, D., … Danish, J. (2020). *Turning Bugs into Learning Opportunities: Understanding Debugging Processes, Perspectives, and Pedagogies*. https://repository.isls.org//handle/1/6661

Kayumova, S., & Dou, R. (2022). Equity and justice in science education: Toward a pluriverse of multiple identities and onto-epistemologies. *Science Education*, *106*(5), 1097–1117. https://doi.org/10.1002/sce.21750

Klahr, D., & Carver, S. M. (1988). Cognitive objectives in a LOGO debugging curriculum: Instruction, learning, and transfer. *Cognitive Psychology, 20*(3), 362–404. https://doi.org/10.1016/0010-0285(88)90004-7

Lafuente Martínez, M., Lévêque, O., Benítez, I., Hardebolle, C., & Zufferey, J. D. (2022). Assessing Computational Thinking: Development and Validation of the Algorithmic Thinking Test for Adults. *Journal of Educational Computing Research*, *60*(6), 1436–1463. https://doi.org/10.1177/07356331211057819

Lather, P., & St. Pierre, E. A. (2013). Post-qualitative research. *International Journal of Qualitative Studies in Education*, *26*(6), 629–633.

Liu, Z., Zhi, R., Hicks, A., & Barnes, T. (2017). Understanding problem solving behavior of 6–8 graders in a debugging game. *Computer Science Education*, *27*(1), 1–29. https://doi.org/10.1080/08993408.2017.1308651

Michaeli, T., & Romeike, R. (2019). Improving Debugging Skills in the Classroom: The Effects of Teaching a Systematic Debugging Process. *Proceedings of the 14th Workshop in Primary and Secondary Computing Education*, 1–7. https://doi.org/10.1145/3361721.3361724

Murris, K. (2020). *Navigating the postqualitative, new materialist and critical posthumanist terrain across disciplines: An introductory guide*. Routledge.

Rich, P. J., Egan, G., & Ellsworth, J. (2019). A Framework for Decomposition in Computational Thinking. *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 416–421. https://doi.org/10.1145/3304221.3319793

Rijke, W. J., Bollen, L., Eysink, T. H., & Tolboom, J. L. (2018). Computational thinking in primary school: An examination of abstraction and decomposition in different age groups. *Informatics in Education*, *17*(1), 77–92.

Rosebery, A. S., Ogonowski, M., DiSchino, M., & Warren, B. (2010). "The Coat Traps All Your Body Heat": Heterogeneity as Fundamental to Learning. *Journal of the Learning Sciences*, *19*(3), 322–357.

Rosiek, J. L., & Gleason, T. (2017). Philosophy in research on teacher education: An onto-ethical turn. *The Sage Handbook of Research on Teacher Education*, 29–48.

Rosiek, J. L., & Snyder, J. (2020). Narrative inquiry and new materialism: Stories as (not necessarily benign) agents. *Qualitative Inquiry*, *26*(10), 1151–1162.

Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a Phenomenology of Computational Thinking in STEM Education. In M. S. Khine (Ed.), *Computational Thinking in the STEM Disciplines* (pp. 49–72). Springer International Publishing. https://doi.org/10.1007/978-3-319-93566-9_4

Sengupta, P., Dickes, A., & Farris, A. V. (2021). *Voicing code in STEM: A dialogical imagination*. MIT Press. https://books.google.com/books?hl=en&lr=&id=cML0DwAAQBAJ&oi=fnd&pg=PR11&dq=Sengupta,+P.,+Dickes,+A.,+%26+Farris,+A.+V.+(In+Press).+Voicing+Code+in+STEM:+A+Dialogical+Imagination.+Cambridge,+MA:+MIT+Press.&ots=KOsR1rV37t&sig=rzTrq0FzDQ8T79IfPjWkJMH2tIk

Shotter, J. (2015). On being dialogical: An ethics of 'attunement.' *Context*, *137*(2015), 8–11.

Silvis, D., Clarke-Midura, J., Shumway, J. F., Lee, V. R., & Mullen, S. (2022). Children caring for robots: Expanding computational thinking frameworks to include a technological ethic of care. *International Journal of Child-Computer Interaction*, *33*, 100491. https://doi.org/10.1016/j.ijcci.2022.100491

Simpson, L. B. (2017). *As we have always done: Indigenous freedom through radical resistance*. U of Minnesota Press.

Sohr, E. R., Gupta, A., Elby, A., & Radoff, J. (2023). How a child entangles empathy and computational thinking in reasoning about fairness. *Early Childhood Research Quarterly*, *65*, 92–101. https://doi.org/10.1016/j.ecresq.2023.05.010

Warren, B., & Rosebery, A. S. (2011). Navigating Interculturality: African American Male Students and the Science Classroom. *Journal of African American Males in Education*, *2*(1).

Warren, B., Vossoughi, S., Rosebery, A. S., Bang, M., & Taylor, E. (2020). Multiple ways of knowing: Re-imagining disciplinary learning. *Handbook of the Cultural Foundations of Learning*, 277–294.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

An Engineering-Focused STEAM Education in Primary Schools: Universal Implementation Strategies

Hau Kwan, CHAN[1], Ka Yuk Cora SIU[2]
[1] St. Patrick's School, Hong Kong SAR
[2] The Education University of Hong Kong, Hong Kong SAR
spschk@sps.edu.hk, corasiu@eduhk.hk

Abstract: *Innovative technology is a key driver of Hong Kong's future development. Hong Kong has actively promoted STEAM (Science, Technology, Engineering, Arts, and Mathematics) education. Since 2016, initiatives such as the Jockey Club Computational Thinking Education Programme (CoolThink@JC) and the Education Bureau's supplementary document "Computational Thinking and Coding Education: Supplementary Document for Primary School Curriculum (Draft)" have supported schools in systematically planning and implementing coding education, emphasizing the importance of innovation and technology education. The authors believe that innovation and technology education should not be equated solely with coding education. To enhance students' practical abilities and ultimately improve overall innovation and technology capabilities, Hong Kong's innovation and technology education must prioritize the significance of engineering ("E" — Engineering). Engineering education should include four essential elements: defining problems, developing prototypes, analyzing data, and reflecting on and presenting findings. The authors suggest placing engineering ("E" — Engineering) at the core of innovation and technology education. Through project-based learning (PBL), students can be nurtured to develop engineering literacy and problem-identification and solving skills. To cultivate students' innovative spirit and align with the national high-quality education development strategy, Hong Kong must universalize innovation and technology education, promote practical application in engineering technologies, and ultimately enhance overall innovation capabilities. This will lay a strong foundation for fostering future technology talent in Hong Kong. The authors share detailed teaching practices in this paper.*

Keywords: STEAM Education, STEAM, Engineering Education

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

以工程為軸心的小學普及創科（STEAM）教育

陳厚君 [1*]，蕭家鈺 [2]
[1,]聖博德學校，香港
[2] 香港教育大學，香港
spschk@sps.edu.hk，corasiu@eduhk.hk

【摘要】 科技是香港未來發展的關鍵引擎。香港積極推動創科（STEAM）教育。自 2016 年透過 Coolthink@JC 及教育局課程文件，協助學校系統性地規劃編程教育。然而，創科教育不應僅限於編程。為培養學生的創科工程素養和尋找及解決問題的能力，本文建議以工程學（“E”── Engineering） 為創科教育核心，透過專題式學習（Project-based based Learning, PBL），在學習過程中為學生提供實踐四個主要工程教育的概念包括問題定義、原型開發、數據分析及反思表達。香港需普及創科教育，促進學生在工程技術上的實踐，最終提升整體創科能力，為香港培養未來科技人才奠定良好基礎。本文作者分享了他的教學實踐細節。

【關鍵字】 創科教育；STEAM；工程教育

1. 前言

創科是推動香港未來發展的重要引擎，近年香港積極推動創科 (STEAM —— 科學、科技、工程、藝術和數學）教育，旨在培養學生的創新精神，落實國家高質教育發展策略。自 2016 年起，賽馬會運算思維教育計畫 (Coolthink@JC) 為香港高小制定課程及提供教師培訓。於同年 12 月教育局於公布《推動 STEM 教育－發揮創意潛能》報告建議在小學階段引入編程來發展學生的計算思維。該報告鼓勵學校提供機會，讓學生透過參與合適的編程活動，學習和應用計算思維和編程技巧並且提升小學生運用創新科技的解難能力。翌年 11 月教育局公布了《計算思維－編程教育：小學課程補充文件（擬訂稿）》供學校參考及採用，以助學校有系統地規劃和推行編程教育為普及創科教育。此外，教育局增設創科教育統籌主任，促進創科教育的發展，奠定其重要性。

特別得益於 Coolthink@JC 及教育局的高小增潤編程教育課程，令小學生對編程教育並不陌生，小學生普遍的動手編程能力已經較之前大大提升。然而，創科教育應不限於編程，亦必須強調當中工程 (“E” —— Engineering) 元素的重要性。工程教育包括四個必要內容：定義問題、開發原型、數據分析及反思表達。此舉可促進學生在工程技術上的實踐，最終提升整體創科能力，為香港培養未來科技人才奠定良好基礎。建議將工程學作為創科教育的核心，通過專題式學習（Project-based Learning, PBL），更能讓學生從小培養創科工程的素養及尋找問題的能力。隨著創科教育在香港的逐步推行，現時的 STEAM 教育發展也在不斷進步和演變。 在這樣的背景下，了解當前 STEAM 教育的發展現狀及其面臨的挑戰和機遇，對於進一步提升教育質量和培養學生的創新能力至關重要。以下將探討香港現時 STEAM 教育的發展情況及其未來方向。

2. 創科(STEAM) 教育現況

近年香港積極推動創科教育。以全方位學習的模式推行。香港教育局鼓勵學校成立核心小組以統籌跨學科的創科教育。一般小學多以常識科或數學科為知識點，輔以電腦科的編程工具如: Scratch, MIT App inventor 2 或 micro:bit 作跨學科專題研習。部份領先的學校會制定

出校本的專題研習，但更多的學校則以各學科為主軸，依靠出版社的教材及內容進行 STEAM 活動。上述兩個方案，在開展創科教育初期無疑是不錯的選擇，也是比較容易達成的選項。

為普及創科教育，教育局於 2022 年增設創科教育 (STEM) 統籌主任，及為資助小學提供聘任彈性以推動創科教育，鼓勵有創科相關知識的人仕協助學校創科教育的發展。可見香港政府對創科教育及培養創科人材的渴望。現時部分學校依賴出版社的教材，並以比賽形式評核學生的能力，形成對創科發展着重資優人才培育的誤解。

一般情況下，學校在 STEAM 領域獲得的獎項數量常被用作評估其 STEAM 教育成效的指標。這現象出現了兩大迷思。其一，只有頂尖的學生才能成為科創人員。其二，學生參加比賽時，是否真的能按步就班地完成所有的學習經歷，還是依靠老師、導師或家長的高度指引或協助下完成？由於每人都成長速度也有不同，假若只有頂尖的學生才能接觸到創科教育，這無疑是扼殺了學生的平等學習機會。這無疑是對學業成績平平但有志成為未來科創人員的學子，築起了一道難以超越的屏障，阻礙他們得到啟蒙的機會和空間。要培養學生的創新精神，落實國家高質教育發展策略，香港需要普及創科教育，促進學生在工程技術上的實踐，最終提升整體創科能力，為香港培養未來科技人才奠定良好基礎。

3. STEAM 教育中的工程

現在的創科教育多以不同的學科作跨學科學習。舉例說：常識科的主題結合電腦科的工具創作出不同的跨學科專題，而在課時的運用，可能是各作分享，例如：需時 6 節課的專題，4 節在常識科的課時，2 節用電腦科的課時。兩個科目皆有其不同的學科目標要達到，所以這樣的跨學科學習面臨著主題上相通，但創作經驗不連貫的問題。如靠單依靠出版社的教材進行的 STEAM 活動則更需要改善。雖然這類的動手經驗也值得支持，但像從前以延伸活動出現比較準確。創科教育的學習應有基本的規範，方便讓學生及家長區分出來，以分辨出學生的熱情與潛能，作中長遠的學習規劃。

小學科學科的成立實在是明智之舉，課程巧妙地將科學與工程從常識的廣闊中抽離，讓老師更專業、專注地將科學與工程的知識及熱情傳遞，但依然值得再進行分拆，達至科學與工程分家，因為科學是以探究為主，而工程則以解決問題為主。雖密不可分，卻有本質的差異（李志民 2024）。經過過去三年的先導階段及增聘統籌主任，現在是一個良好契機去優化本港創科教育，特別是在工程（“E” —— Engineering）及科技（“T” —— Technology）方面。所以建議將以工程學為主的 STEAM 獨立成科，可讓學生從小培養出創科工程的素養及尋找問題的能力。以專題式學習為教學法推進，讓每一位學生都可親自嘗試解決生活中的不同問題，如工程師的思維去學習。

3.1 工程獨有的內容

Simarro & Couso (2021) 定義了九個工程的步驟，包括 一. 定義和界定工程問題 ；二. 開發和使用原型和模擬；三. 規劃並進行測試 ；四. 分析和解釋數據以確定改進點；五.運用數學和計算思維、科學模型和現有技術；六. 識別和/或開發多種解決方案並選擇最佳的一種；七. 實現解決方案；八、有據可依；和九. 獲取、評估和交流訊息。本文根據小學科學科對工程設計的必要步驟而提出一個四步曲的簡化版 DDAR:

3.1.1 定義及界定工程問題 （Defining and delimiting engineering problems）

第一步最為重要，卻又常被忽視。在現有的課堂中，老師會提出今天的主題或引入難題的情境，但往往因為學生缺乏對問題作深入磨研。例如:運用五何法，了解到問題對社會什麼人地事時做成問題，並且協助學生多了解工程的發展如何改善人們的生活或令世界更為公平，讓他們看到工程的美和價值。因此，老師在引導學生深入理解問題的同時，也需要幫助他們

明確界定成功的準則。界定問題也包括如何制定成功準則，如果沒有成功準則，或讓成功準則模糊不清，學生在自主推進項目時，對方向掌握及時間、資源或資金的安排會欠缺經歷。

3.1.2 開發及製作原型作模擬 （Developing and using prototypes and simulations）

這是學生們最為激動的階段，他們急切希望開始動手製作。然而，他們往往忽略了一個工程中至關重要的元素——開發過程。若在沒有適當規劃的情況下立即進行項目的構建，將可能導致材料、人力和時間的浪費。因此，在開始製作之前，必須先完成詳細的工程設計，以避免資源的浪費。對於小學生而言，重點不應放在嚴格的格式或精確度上，而應著重於對概念的理解及良好習慣的培養。教師可以鼓勵學生展示各種設計圖，例如機械工程師所用的正交視圖、電機工程師常見的電路圖，或電子電腦工程師經常使用的流程圖。

透過參與初步的開發階段，學生在製作原型時更有可能產生多樣性的作品，這是一個值得讚賞的結果。然而，當他們進入原型製作階段時，必須運用來自各工程學科的不同技術，特別是 STEAM（科學、技術、工程、藝術、數學）中的技術層面。這不僅包括學生的編程能力，還涵蓋了適用於人類各項工程活動的多種工程技能。例如，機械工程可能涉及螺絲和不同齒輪組合的使用，而電機/電子工程則常用電池、鱷魚夾、麵包板及杜邦線等工具。這些都是小學生能夠操作的基本工具。因此，教師在課堂中提供指導至關重要；否則，學生的作品可能僅淪為單純的視覺藝術作品，或是只能依賴教材內容進行製作，從而限制了他們的創造力。

3.1.3 分析數據以改良產品 (Analyzing and interpreting data to identify points for improvement)

根據成功準則作分析及評鑑產品的好與壞，既與〝第一步：定義及界定工程問題〞緊扣，也同樣是時侯修訂成功準則而優化項目。一般小學生只能按項目的外觀給予意見，如果老師能提醒與成功準則的內在關係，學生定能更好的給出準確的評鑑。數據的指標更應向學生介紹，能讓成功準則更加準確及具體。在改良項目時，可重覆〝第二步：開發及製作原型作模擬〞的流程，先由設計圖開始作修改，再修改實物。鼓勵學生將設計圖及思考過程有系統地記錄及保存。透過這些記錄看到自己的進步，也增強他們堅毅的精神。

3.3.4 表達和反思 (Representation and Reflection)

準備項目報告及展示，多樣化的結果可讓學生與學生能有不同的衝擊及價值，在同一的主題下，由設計到原型到成品，都能看到相似及不同，可令學生覺得新鮮，更擁有成就感。完成展示後，也可邀請同學繼續改良產品。如以小組形式進行項目，可以鼓勵學生間互相協作及共同成長。

4. 校本課程設計與實施

如要達成以上以工程為主的科技創新教育跨學科課程，學校需要訂明一個獨立的課時，有規範地進行，並透過 STEAM 統籌主任協調不同的學科配合進行，以下是筆者學校的做法。

4.1 課時的製造

根據指引，小一至小六的課程中，電腦科每循環週設有一節 35 分鐘的課堂。我校在初小階段維持每循環週一節的安排；然而，高小的電腦科則改為以 STEAM 活動形式呈現。為確保每位學生都能接觸到工程學的基本概念，並有機會學習以工程為本的 STEAM 內容，高小課程安排為每循環週兩節連堂。課時的調整自然需要課程內容適度的取捨，例如，原電腦科所教授的 Microsoft 文書軟件，如 PowerPoint 等內容，將不再於課堂內講授，而是改為在其他學科的專題中以自學形式進行。此外，MS Word 的學習將與英文鍵盤操作相結合，而 MS Excel 則併入函式的相關學習中。基本中文輸入法（如倉頡和速成）則改為教授普通話拼音輸入，並融入普通話科中。基本的電腦操作對當代學生仍然重要，但這已成為他們近乎與生俱來的

能力。而隨著 iPad 的直觀操作普及，電腦科得以釋放更多課程空間，用於工程相關內容的發展。

4.2 動手做的學習框架

課程的安排需要有系統、循序漸進。自高小四年級起，我校將 micro:bit 納入課程，相信這也是大部分學校的共同選擇。然而，我校著重加強學生工程思維的發展，因此在課程設計上，不僅提高了編程的難度，也加強了電子電機工程技術的學習。小四課程設計重點在於 micro:bit 的原有功能、內建感應器和接腳，再加上學習使用杜邦線及鱷魚夾控制 LED。小五，學生將運用帶有電機的擴展板，學習控制直流電機的運作。最後，在小六階段，學生進一步使用擴展板及麵包板來控制伺服馬達。通過這樣的設計，學生能逐步接觸並操作多種技術，為未來的工程學習打下堅實基礎。

4.3 跨學科的協作

由學校的電子學習小組改變成為 STEAM 小組，除了分享學校 BYOD 的情況外，在這會議中也邀請不同學科合作，由 STEAM 科作工程的製作，而其他科目作應用。例如：製作樂器在 STEAM 負責由整個製作，在音樂課時一起應用。一方面減少了其他科老師從 STEAM 科的壓力，也提供了機會予學生應用所學過跨學科的知識，達到科技解決問題的工能。

5. 總結

從小學開始啟動以工程為主的創科教育，有望提升學生整體的能力，為香港科技創新培養良好的土壤以及更多優秀人材。要在學校實現這種以工程為核心的創科教育模式，首要條件是在課程中劃出明確且獨立的課時進行規範化安排。具體來說，要將這部分課程改為 STEAM 活動，學校需要妥善安排課堂時間，建議每週兩堂連續課確保每位學生能夠接觸並理解工程學的基本概念。課程的內容亦需要進行調整。例如，一些關於電腦基本運作可考慮與其他學科合併或作專題自學，例如普通話拼音輸入可以融入普通話科目之中。這樣的調整不僅能騰出更多課程空間來教授工程相關內容，也能讓學生獲得多元化的學習經驗。此外，STEAM 學習活動的設計需要一個系統且逐步深入的動手學習框架。逐步提升課程中的難度，這種按年級層層遞進的設計，旨在建立學生對工程思維和電子技術的扎實基礎，讓技術學習更加深入。成功的 STEAM 教育還需要跨學科的協作來達成。學校可以設立創科教育核心小組，並邀請各學科教師共同參與，以減輕其他科目教師的負擔，還讓學生能夠通過實際應用學到的知識，體驗跨學科協作如何解決實際問題。透過上述方法，學校可以系統性地將工程為核心的創科教育融入課程之中，幫助學生培養創新能力，強化解決問題的實踐技能，為未來科技創新打下堅實的基礎。

参考文獻

教育局（2015)。**推動 STEM 教育－發揮創意潛能**。課程發展議會

教育局（2020)。**計算思維－編程教育：小學課程補充文件（擬訂稿）**。課程發展議會

李志民（1998）。**科學與工程的區別**。人才發展專業委員會

Simarro, C., & Couso, D. (2021). Engineering practices as a framework for STEM education: a proposal based on epistemic nuances. *International Journal of STEM Education, 8*(1), 53.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

A Programming Learning Platform with Misconception Diagnosis

Si-Ru CHEN[1*], Yu-Tzu LIN[2]
[1,2]Graduate Institute of Information and Computer Education, National Taiwan Normal University, Taiwan
81208002E@ntnu.edu.tw, linyt@ntnu.edu.tw

Abstract: *Programming novices often encounter difficulties in mastering fundamental concepts due to common misconceptions, which can manifest as syntax errors, logical flaws, and incorrect problem-solving strategies. To address this challenge, this study develops a programming learning platform that provides real-time diagnosis and feedback on misconceptions. Students can submit code to verify its correctness and receive immediate feedback on identified misconceptions, while instructors can monitor learning progress and adjust instructional strategies accordingly. The platform utilizes data mining techniques to extract code features, which are then analyzed using spectral clustering algorithms to identify misconception symptoms and categorize them into specific types. By mapping the relationships among various misconceptions, the platform enhances instructors' understanding of students' learning obstacles. Future work will focus on designing more guided feedback based on misconception patterns to better support students in correcting misunderstandings and improving their programming skills.*

Keywords: Misconception diagnosis, Programming instruction, Data mining

1. Introduction

Novice programmers often face learning challenges, attributed not only to the complexity of programming concepts but also to persistent misconceptions that impede understanding (Durkin & Rittle-Johnson, 2015; Kaczmarczyk et al., 2010; Qian & Lehman, 2017). Although previous studies have developed diagnostic tools to identify misconceptions (Chen et al., 2017), few have incorporated real-time automation within learning environments. With the emergence of data mining techniques in education, new opportunities have arisen for diagnosing learning difficulties more effectively (Ihantola et al., 2015; Steven, 2014). Building on machine learning–based diagnosis frameworks (Lin et al., 2022), this study proposes a platform that integrates real-time misconception detection, allowing instructors to continuously monitor student progress and intervene promptly (Robinson & Carroll, 2017; Watanobe et al., 2020).

2. Methodology

Code features were collected from 1,213 novice programmers from high schools and universities in Taiwan, each possessing 6 to 12 months of programming experience. All participants completed a misconception diagnosis test designed to be language-independent.

For the clustering of code features, a two-stage spectral clustering approach was employed to group feature vectors derived from student submissions associated with misconceptions. This method enabled the identification of latent patterns within the data and facilitated the mapping of clusters to corresponding misconceptions. Within each cluster, features that appeared in more than 30% of the submissions were designated as symptoms of misconceptions.

3. Results

3.1. Symptoms of Programming Misconceptions

To successfully diagnose students' programming misconceptions and provide targeted feedback within the learning platform, we defined symptoms of programming misconceptions (Table 1), serving as indicators that allow the system to identify potential misunderstandings in students' code submissions and support timely instructional interventions.

Table 1. List of Symptoms of Programming Misconceptions.

Misconception Identifier	Symptoms of Code	Example
M1.1.1	Using a *while* statement for a single condition check or an *if* statement for multiple condition checks.	Program 1: *x = 1* *if x == 1:* *print(x)* Program 2: *x = 1* *while (x == 1):* *print(x)* Misconception: Believes *if* and *while* are functionally equivalent, both executing once for condition checking.
M2.1.1	The loop control variable is assumed to start from 0, with the condition set as <= or <, and the increment always by 1.	*a = 5* *while (a > 1):* *a = a – 1* *print(a)* Misconception: The variable *a* starts at 0 and increases by 1 each time the loop runs. The loop exits when *a > 1*. Therefore, the loop runs three times. Step 1: *a* = 0 → 0 - 1 = -1 → not greater than 1 → loop continues. Step 2: *a* = 1 → 1 - 1 = 0 → not greater than 1 → loop continues. Step 3: *a* = 2 → 2 - 1 = 1 → not greater than 1 → loop continues. Step 4: *a* = 3 → 3 - 1 = 2 → 2 > 1 → loop exits.
M2.1.2	The loop body only references the control variable.	*y = 5* *z = 0* *for x in range(0, y, 1):* ____________________ *print(z)* Misconception: The variable *y* should be updated as *y = y + 1* because only the variables used in the loop condition are relevant inside the loop.
M2.2.2	Using an *"if"* statement instead of a loop for repeated execution.	*int x = 0;* *if (x > 0; x++) { cin >> x; cout << x!; }* Misconception: Use of *if* statements instead of loops.
M3.1.1	The output results in one iteration fewer than expected.	*int a = 5;* *while (a > 1){ a = a - 1; }* *printf("%d", a);* Misconception: The output is 4,3,2 because the condition requires *a* to be greater than 1, so the loop stops at 2.
M3.1.2	In a variable swap task, using only one or two assignment statements and omitting a temporary variable.	*int x;* *while (y) { x = y; }* Misconception: Swapping two variables can be done directly without using a temporary variable.
M3.1.3	Determining the execution count based on loop body statements rather than loop conditions.	*x = 0* *while (x < 3):* *x = x + 1* *print(x)* Misconception: The statement *x = x + 1* is executed twice: first, 0 + 1 = 1; then, 1 + 1 = 2. Therefore, the loop executes twice.
M3.2.1	Adding a condition after *"else"*.	*int x;* *if (x == 1){ cout("Congratulations, you got the ticket");* *} else (x == 0){ cout("Unfortunately, no tickets left"); }* Misconception: Learner incorrectly assumes *else* requires a condition.

M4.1.1	Parallel loop statements at the same level.	*for (a = 10; a < 20; a++){ b = b - 1;}* *for (a = 0; a < 10; a++){ b = b + 1;}* Misconception: If *a* starts at 0, the second loop executes first. The first loop will only execute when *a* reach 10.
M4.2.1	Parallel conditional statements at the same level.	*int a = 1;* *if (a == 0){ cout("zero"); }* *if (a == 1){ cout("one"); }* Misconception: *a* is 1, it directly jumps to *if(a == 1)*.

3.2. Clustering of Programming Misconceptions

To enhance the effectiveness of the platform's feedback in addressing different types of programming misconceptions, a clustering process was conducted to categorize common misconception patterns. The clustering results identified a total of 11 distinct clusters, each corresponding to a specific type of programming misconception (Table 2). These clusters form the basis for mapping student errors to targeted feedback strategies within the platform.

Table 2. Clustering Results from Data Mining and Correspondence with Programming Misconceptions.

Cluster ID	Corresponding Programming Misconceptions	Cluster ID	Corresponding Programming Misconceptions	Cluster ID	Corresponding Programming Misconceptions
Cluster 1	M2.1.2. M3.1.1.	Cluster 5	M3.1.1. M4.2.1.	Cluster 9	M3.2.1.
Cluster 2	M3.1.1. M3.1.2.	Cluster 6	M2.1.1.	Cluster 10	M4.1.1.
Cluster 3	M3.1.1. M3.2.1.	Cluster 7	M2.1.2.	Cluster 11	M4.2.1.
Cluster 4	M3.1.1. M4.1.1.	Cluster 8	M3.1.2.		

3.3. Programming Learning Platform with Misconception Diagnosis

The programming learning platform developed in this study diagnoses student misconceptions based on clustering results derived from code feature analysis. Learners can write and submit code directly through the platform, which automatically evaluates the correctness of their solutions and detects potential misconceptions. Instructors are provided with a backend interface that allows them to create new programming problems, edit existing ones, and specify the misconceptions to be diagnosed for each problem. Additionally, instructors can define new misconception types and corresponding feedback messages, supporting the continuous expansion and refinement of the diagnostic framework. All student submissions, along with the identified misconceptions, are systematically recorded and made available for instructor review and monitoring. The platform's user and instructor interfaces are illustrated in Figures 1 and 2.

Figure 1. Homepage and Student Submission Interface.

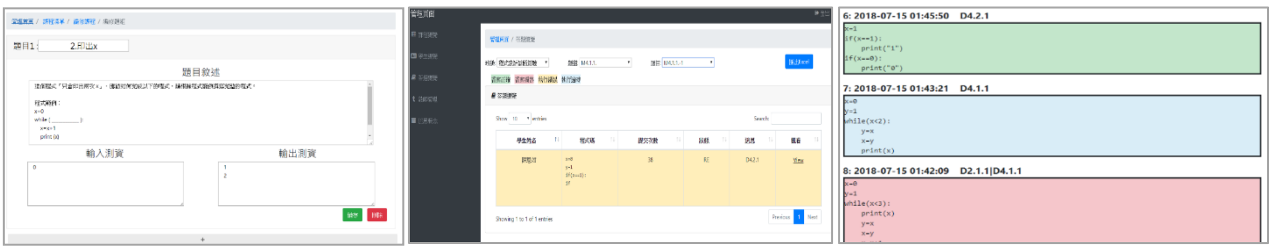


Figure 2. Instructor Interface with Student Submission Records and Misconception Diagnosis Results.

4. Discussion and Conclusions

4.1. Establishing the Correspondence Between Novice Programming Misconceptions and Syntactic Symptoms

The study identified 11 misconception clusters through data mining, each representing a distinct type of programming misconception. Among these, one cluster (M3.1.1) demonstrated a particularly broad influence on students' understanding. It reflects the misconception that students determine the number of loop executions solely based on the upper limit value, rather than evaluating the loop's conditional expressions to control the flow of execution. This misconception spans both conditional evaluation and control flow concepts, and thus can propagate misunderstandings into other related areas of programming logic. Immediate feedback is crucial to correct such fundamental misconceptions.

4.2. Design and Implementation of an Automated Diagnosis Mechanism for Programming Misconceptions

The programming learning platform developed in this study automatically diagnoses misconceptions at the point of student code submission. Diagnosis results are recorded in the platform, enabling teachers to monitor students' misconceptions and track changes in their understanding over time. Future research could further investigate the design of guided feedback mechanisms, aiming to assist students in correcting misconceptions more effectively and promoting deeper conceptual understanding.

References

Chen, S.-R., Lin, Y.-T., & Wu, C.-C. (2017). The development of the diagnostic inventory of programming misconceptions. *International Conference on Education and Technology (ICET 2017)*, 32.

Durkin, K., & Rittle-Johnson, B. (2015). Diagnosing misconceptions: Revealing changing decimal fraction knowledge. *Learning and Instruction*, *37*, 21–29.

Ihantola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S. H., Isohanni, E., Korhonen, A., Petersen, A., Rivers, K., Rubio, M. Á., Sheard, J., Skupas, B., Spacco, J., Szabo, C., & Toll, D. (2015). Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies. *Proceedings of the 2015 ITiCSE on Working Group Reports*, 41–63. https://doi.org/10.1145/2858796.2858798

Kaczmarczyk, L. C., Petrick, E. R., East, J. P., & Herman, G. L. (2010). Identifying student misconceptions of programming. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education - SIGCSE '10*, 107. https://doi.org/10.1145/1734263.1734299

Lin, M.-H., Chen, S.-R., Liao, J., & Lin, Y.-T. (2022). Automated Diagnosis of Novice Programmers' Misconceptions Based on Machine Learning. *Conference Proceedings (English Paper) of the 26th Global Chinese Conference on Computers in Education (GCCCE 2022).*, 53–60.

Qian, Y., & Lehman, J. (2017). Students' Misconceptions and Other Difficulties in Introductory Programming. *ACM Transactions on Computing Education*, *18*(1), 1–24. https://doi.org/10.1145/3077618

Robinson, P. E., & Carroll, J. (2017). An online learning platform for teaching, learning, and assessment of programming. *2017 IEEE Global Engineering Education Conference (EDUCON)*, 547–556.

Steven, F. (2014). Predictive Analytics, Data Mining and Big Data. Retrieved from https://link.springer.com/content/pdf/10.1057%2F9781137379283.pdf

Watanobe, Y., Intisar, C., Cortez, R., & Vazhenin, A. (2020). Next-Generation Programming Learning Platform: Architecture and Challenges. *SHS Web of Conferences*, *77*, 01004. https://doi.org/10.1051/shsconf/20207701004

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Developing Computational Thinking Through Interactive Storytelling in English Teacher Education

Hsin-I CHEN
National Taipei University of Technology, Taiwan
hichen@ntut.edu.tw

Abstract: *This study investigates how pre- and in-service English teachers develop and apply computational thinking (CT) competencies through interactive digital storytelling projects. Responding to research gaps at the intersection of CT, L2 literacy education, digital literacy, and teacher education, the study examines how CT can enhance digital and multimodal literacies in English teaching contexts. Data were collected through screen recordings of their design processes, pre- and post-intervention questionnaires, and semi-structured interviews. Qualitative analyses of screen recordings revealed active engagement with CT competencies. The findings highlight how digital storytelling fosters CT skill development and creative literacy practices, positioning teachers as content designers. The study offers pedagogical implications for teacher education, advocating for CT-integrated training that fosters CT competency, digital literacy, and innovative L2 pedagogy in technology-rich learning environments.*

Keywords: Computational Thinking, Interactive Storytelling, Literacy Education, Teacher Education

1. Introduction

In the context of 21st-century education, computational thinking (CT) has become a crucial competency not only for STEM educators but also for language teachers. As education systems increasingly integrate digital technologies and emphasize multimodal literacies, CT provides language teachers with essential problem-solving strategies and instructional design skills (Li et al., 2024; Jacob & Warschauer, 2018).

CT fosters essential skills that enhance language learning by promoting higher-order thinking, such as planning, analyzing, and organizing ideas. Integrating CT into English writing pedagogy improves learners' ability to generate and structure ideas, supporting writing fluency and coherence (Wu et al., 2024). CT also empowers language teachers to create technology-enhanced learning environments through digital storytelling, gamification, and interactive activities, which boost learner engagement and digital literacy (Yu et al., 2024). Additionally, CT strengthens teachers' digital competence and 21st-century skills, helping them adapt to evolving educational technologies (Nouri et al., 2020).

While CT has gained attention in education, its integration into second language (L2) education remains underexplored. Yu et al. (2024) highlight the lack of empirical studies investigating how CT skills such as abstraction, decomposition, and algorithmic thinking can be effectively incorporated into language learning curricula, particularly in L2 contexts. Many pre-service and in-service language teachers lack sufficient training in digital literacy and computational thinking. Specifically, Parsazadeh et al. (2021) highlight that research on integrating CT into digital storytelling (DST) for enhancing L2 learners' motivation and achievement is still in its early stages. They emphasize the need for more comprehensive empirical studies to examine the pedagogical impact of combining CT and DST within L2 learning contexts.

In response to the identified research gaps, this study investigates how pre-service English language teachers develop and apply CT competencies through the creation of interactive DST projects. Specifically, it addresses the limited empirical research on integrating CT into L2 literacy practices and explores how CT can enhance digital and multimodal literacies within language teaching and learning contexts (Jacob & Warschauer, 2018). The central research question guiding this study is: How do pre-service English language teachers demonstrate computational thinking competencies during the process of interactive digital storytelling design?

2. Methodology

This study involved 16 pre- and in-service English teachers participating in a professional development program focused on enhancing their CT competencies and digital literacy. Using a mobile-based visual programming platform, the participants designed interactive storybook tasks for future L2 learners. Data were collected through screen recordings, pre- and post-intervention questionnaires, and semi-structured interviews. The screen recording video data from preservice English teachers' interactive storytelling creation process were analyzed through the lens of Atmatzidou and Demetriadis (2016) which provides a structured and evidence-based way to assess their CT competencies. Since they define CT as a set of programming skills/competencies—abstraction, generalization, algorithmic thinking, modularity, and decomposition. This study used these five dimensions to build an analytic framework for the video data. Based on this analytical framework, a detailed coding scheme and codebook were developed, including precise indicators and observable behaviors aligned with each CT competency. For example, abstraction was identified when participants simplified complex narratives by focusing on core characters or plot elements and deliberately excluding extraneous details. Generalization was evident when code blocks or storytelling structures were reused for different characters or narrative scenes. Algorithmic thinking was observed in participants' ability to sequence actions logically, utilizing event triggers or incorporating loops to automate repetitive actions. Modularity was demonstrated through the creation of distinct blocks for characters, dialogues, or specific actions, with logical naming conventions that facilitated reuse. Finally, decomposition was noted when participants divided the overall project into discrete tasks, such as designing characters, coding movements, or scripting dialogues. The coding schemes were summarized in Table 1. The screen recordings were segmented into analyzable units based on time intervals, participant actions, and task delineations, providing a systematic basis for examining the application of CT competencies during the storytelling creation process.

Table 1. Indicators and observable behaviors for each CT competency in the screen recordings

CT Competency	Observable Behaviors (Examples from Screen Recordings)
Abstraction	Simplifying complex stories by focusing on key characters or plot events; ignoring unnecessary details.
Generalization	Reusing code blocks or storytelling structures for different characters or scenes.
Algorithmic Thinking	Sequencing actions logically (e.g., event triggers); creating loops.
Modularity	Creating separate blocks for characters, dialogue, or actions; naming code sections logically.
Decomposition	Breaking the project into tasks: e.g., designing characters, coding movements, scripting dialogues.

3. Findings

The analysis of screen recording data indicated that both pre-service and in-service EFL teachers actively applied core CT competencies during their interactive storytelling projects. One frequently observed practice was decomposition, as participants systematically broke down complex narratives into smaller, programmable segments. For instance, when adapting Le Petit Prince, teachers organized the storyline into separate chapters, each corresponding to a different planet. Within these divisions, they designed interactive tasks that encouraged learners to engage with thematic vocabulary and dialogue. In addition, modularity played a key role in their design processes. Teachers constructed reusable coding blocks for characters and dialogues, which streamlined the development of consistent interactions across the story. This was particularly evident in their version of Brown Bear, Brown Bear, What Do You See?, where they created modular components for each animal character. This method not only simplified the coding process but also maintained narrative coherence and improved learner interaction.

Furthermore, algorithmic thinking surfaced in the logical arrangement of actions. For example, participants programmed event triggers that prompted learners to click on a character—such as the fox in Le Petit Prince—to activate vocabulary prompts or audio narration. Teachers demonstrated abstraction by reducing narrative complexity, focusing on essential plot events and language features to aid comprehension for L2 learners. Generalization was evident as they reused interaction scripts across various story scenes and adapted them for other language learning tasks, including matching exercises and comprehension checks.

Across all data sources, it became clear that participants' work with interactive storytelling, particularly through the lens of well-known children's literature, fostered a new dimension of computational literacy. By designing story-driven learning tasks that merged language pedagogy with CT competencies, teachers adopted the role of creative content developers, moving beyond the traditional textbook-based approach. Their adaptations of Le Petit Prince and Brown Bear, Brown Bear, What Do You See? showcased how computational thinking, digital storytelling, and L2 literacy instruction can be effectively intertwined. The participants expressed a heightened awareness of how computational practices, such as decomposition and modularity, paralleled traditional literacy strategies, such as outlining and text structuring, reinforcing the argument by Jacob & Warschauer (2018) that CT should be considered a fundamental literacy in digital learning environments (Duran, 2025).

4. Conclusion & Implication

This study explored how pre- and in-service English language teachers applied CT competencies—such as abstraction, algorithmic thinking, and decomposition—while designing interactive storytelling tasks. The findings show that participants not only demonstrated CT skills but also engaged in creative literacy practices, aligning with Jacob and Warschauer's (2018) view of CT as an essential literacy. By integrating CT and digital storytelling, teachers shifted from technology consumers to designers of multimodal narratives that support L2 literacy development. These findings foreground important implications for language teacher professional development. Training programs should connect CT principles to language teaching, offer hands-on practice with digital storytelling tools, and encourage reflection on CT's role in fostering digital literacy. Embracing CT as both a cognitive skill and literacy practice can help language teachers create more engaging, inclusive, and future-ready classrooms.

References

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, *75*, 661–670. https://doi.org/10.1016/j.robot.2015.10.015

Duran, V. (2025). Computational thinking in computer-assisted language learning: New directions. In *Handbook of CALL Research* (pp. 117–132). Springer. https://doi.org/10.1007/978-981-96-0853-9_6

Jacob, S. R., & Warschauer, M. (2018). Computational thinking and literacy. *Journal of Adolescent & Adult Literacy*, *61*(4), 433–439. https://doi.org/10.1002/jaal.679

Li, X., Sang, G., Valcke, M., & van Braak, J. (2024). Computational thinking integrated into the English language curriculum in primary education: A systematic review. *Education and Information Technologies*, *29*, 723–746. https://doi.org/10.1007/s10639-024-12522-4

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence, and 21st-century skills when learning programming in K-9. *Education Inquiry*, *11*(1), 1–17. https://doi.org/10.1080/20004508.2019.1627844

Parsazadeh, N., Ali, R., Rezaei, A., & Saeed, N. (2021). Integrating computational thinking into digital storytelling: A systematic review. *Journal of Educational Computing Research*, *59*(7), 1403–1430. https://doi.org/10.1177/0735633120967315

Wu, T. T., Silitonga, L. M., & Murti, A. T. (2024). Enhancing English writing and higher-order thinking skills through computational thinking. *Computers & Education*, *206*, 104621. https://doi.org/10.1016/j.compedu.2024.104621

Yu, X., Soto-Varela, R., & Gutiérrez-García, M. Á. (2024). How to learn and teach a foreign language through computational thinking: Suggestions based on a systematic review. *Thinking Skills and Creativity*, *51*, 101247. https://doi.org/10.1016/j.tsc.2024.101247

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Design and Implementation of an Auto Marking System for MIT App Inventor Coding Education – An Alternative Approach

Jiangshan SUN[1], Shanshan CHEN[2*]

[1,2]South China Normal University, Guangzhou, China

[1]Dipont Huayao Collegiate School Kunshan, Suzhou, China

[2]Canadian International School, Hong Kong SAR

sunjiangshan@hotmail.com, s56223896@gmail.com

Abstract: *This analysis investigates the application of an Auto Marking System (AMS) in primary school Information Technology (IT) education, focusing on MIT App Inventor. Acknowledging the limitations of conventional manual grading, the work delves into AMS as a means to enhance the evaluation process. It presents a well-organized method for AMS adoption and emphasizes major difficulties in assessing App Inventor coding tasks. For educators, AMS significantly reduces grading time, while students gain from immediate feedback and tailored educational experiences. Furthermore, the paper discusses challenges in automated assessment and considers possible advancements in this evolving domain.*

Keywords: Auto Marking System, MIT App Inventor, Coding Education, Elementary Education, Programming Language Processing

1. Introduction

Studies across multiple higher education institutions have shown that grading in IT courses consumes up to 30% of instructors' time, while suffering from inefficiency, subjectivity, and delayed feedback (Paiva et al., 2022). This grading challenge has intensified in recent years across all educational levels, with particular complexity emerging in contexts where visual programming tools like MIT App Inventor are used, as these increase the difficulty of assessment due to the visual and interactive nature of student submissions. Research indicates that automated assessment must extend beyond verifying functional correctness to incorporate multidimensional metrics such as code efficiency, readability, and logical complexity (Paiva et al., 2022). These metrics enable instructors to evaluate critical aspects of student learning, such as algorithmic thinking and code organization skills, which are essential indicators of programming proficiency. However, current systems predominantly use statistical features and rule-based matching, limiting them to surface-level code analysis. This fundamental limitation prevents a deep understanding of code semantics, especially when evaluating open-ended or project-based assignments, where assessing students' creativity and problem-solving skills requires more sophisticated analysis (Ramesh & Sanampudi, 2022). Furthermore, while educational platforms like Scratch or App Inventor successfully engage students through gamification, current automated assessment methods struggle to evaluate the resulting work effectively. Specifically, these tools' rule-based metrics cannot adequately capture higher-order skills such as computational thinking and logical reasoning, which are essential learning outcomes in programming education (Martínez-Murciano & Pérez-Jorge, 2024; Pérez-Jorge & Martínez-Murciano, 2022).

This paper examines the application of Auto Marking Systems (AMS) in coding education. Traditional grading methods place considerable burdens on educators, leading to inconsistencies that compromise fairness and fail to meet individual learning needs. By streamlining the evaluation process, AMS ensures consistency, boosts efficiency, and delivers immediate feedback. In the context of MIT App Inventor, this paper demonstrates how automated assessment can improve elementary-level IT education.

2. Construction of coding file

This section examines the architecture of MIT App Inventor project files (aia), which form the foundation for implementing the Auto Marking System (AMS). Functionally, aia files operate as compressed archives that bundle two core components—scm and bky files—alongside supplementary media attachments (e.g., images, audio). Upon extraction, these files are organized into a unified directory without altering their native formats. The scm files, encoded in JSON, store the program's functional logic, such as event handlers and algorithmic workflows. Conversely, bky files utilize XHTML markup to define visual interface elements, including buttons, layouts, and interactive widgets. By parsing these components, the AMS evaluates both technical execution (e.g., logic accuracy) and design coherence (e.g., UI usability), enabling holistic assessment of student submissions.

3. Design of AMS

The advancement of the AMS calls for a well-rounded strategy, incorporating four key elements. First, it involves tackling issues linked to AI, ensuring its use corresponds with our goals. Second, defining effective marking criteria is essential for precise assessment and evaluation. Third, choosing a suitable coding language is important for optimizing performance and compatibility. Lastly, algorithmic programming forms the backbone of structuring and enhancing the system's functionality. Together, these elements contribute to the successful progress of AMS.

3.1. A realistic consideration—AI or not AI

Our team considered AI as an option, but it didn't meet our expectations. While it had certain advantages, its drawbacks outweighed the benefits in our case. Consequently, we chose to explore other methods.

Table 1. Scenario Comparison.

	Traditional Rule-based Scoring	**Distilled Model AI**
Question Type Complexity	Simple types (e.g., multiple-choice, fill-in-the-blank)	Complex types (e.g., essays, open-ended responses)
Scoring Criteria	Explicit, structured rules	Ambiguous, requires semantic understanding
Data Requirements	No training data needed	Requires large amounts of labeled data
Hardware Resources	Low-end servers	Medium/high-performance servers (GPU required)
Privacy Requirements	No special requirements	Prioritizes localized deployment

Traditional rule-based scoring is more suitable for local schools due to its lower costs, minimal hardware requirements, and ease of maintenance by general IT technicians, which makes it well-suited for standardized assessments.

3.2. Marking Criteria

This involves categorizing coding assignments into task- oriented and project-oriented problems by their distinct evaluation parameters. Task-based problems are evaluated based on predefined solutions, whereas project- oriented problems require innovative and individualized assessments. This essay focuses exclusively on task- based problems.

3.3. Selection of Coding Languages

The AMS is optimized for operation on Windows systems, prioritizing languages natively supported without additional installations. Our previously research using Batch (BAT) plus Visual Basic for Applications (VBA), which are preferred for their innate compatibility with Windows and Microsoft Office. This article tried to use an alternative approach through Python Packing. It generates an executable file to run the coding individually on any device without Python environment.

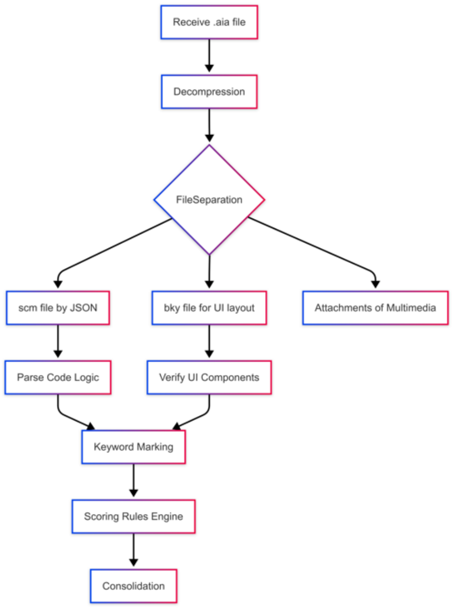


Figure 1. The workflow of AMS

3.4. Algorithm Programming

The AMS establishes a comprehensive marking framework to efficiently address task-oriented coding assignments. This framework is essential for the systematic evaluation of coding tasks that may not conform to a standard structure. The operational methodology of the AMS is delineated in the following steps:

3.4.1. Data Extracting

Utilize compression software to extract the contents of the aia file, employing a method that omits subfolders to facilitate a streamlined review process. The extracted files are then displayed in the current directory for convenient access and analysis.

3.4.2. Keywords Counting and Matching

Develop an algorithm to detect and count specific coding elements within submissions, enabling the AMS to not only confirm the presence or absence of required block types but also to assess the overall coding structure.

3.4.3. Marks Assigning

Ensure the marking process conforms to the educational institution's standards. To perform an in-depth evaluation, multiple code elements may be required to satisfy a single criterion. The AMS assigns marks to each relevant element and combines them to determine the final score for each task.

4. Testing

To evaluate the time efficiency of traditional manual marking, we divided the process into three phases: Pre-treatment, Evaluation, and Documentation, measuring the time spent in each. The same phased approach was then applied to AMS, allowing for a direct comparison of time consumption between the two methods.

4.1. Pre-treatment

This phase involved preparing the aia files for assessment, ensuring they were in the correct format and be able to upload to App Inventor web complier.

4.2. Evaluation

The core of the testing process, this phase, involved evaluating a wide variety of aia files, demonstrating different levels of coding complexity and component selection accuracy. The evaluation criteria included precision in marking and efficiency in processing time.

4.3. Documentations

In this phase, the outcomes of the AMS evaluations were documented, focusing on performance metrics such as marking accuracy, time efficiency, and error identification.

5. Results

A group of 33 students was selected for evaluation using both techniques. The outcomes highlighted a remarkable improvement in accuracy and consistency with AMS over conventional manual marking. AMS showed exceptional precision in identifying correct and incorrect coding components, while reducing assessment time by about 81%. Additionally, its intuitive layout and enhanced design made the grading process easier for instructors to manage.

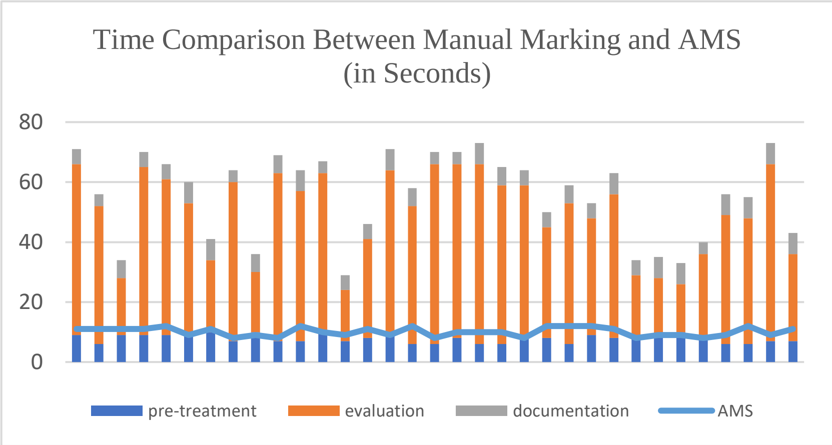


Figure 2. The Time Comparison Between Manual Marking and AMS (in Seconds)

6. Conclusion

The analysis concludes that using an Auto Marking System (AMS) in MIT App Inventor coding classes in primary schools significantly improves the evaluation process. AMS not only boosts efficiency but also provides prompt, detailed feedback, creating a more interactive and flexible learning environment. While the system shows considerable promise, the study points out certain challenges, such as its reliance on fixed criteria and the necessity for regular updates to stay aligned with changing coding standards. Future studies plan to integrate artificial intelligence to enhance the grading system and expand AMS's use to additional programming languages and educational levels. This analysis illustrates the impact of technology in transforming educational assessment practices.

References

Paiva, J. C., Leal, J. P., & Figueira, Á. (2022). Automated assessment in computer science education: A state-of-the-art review. *ACM Transactions on Computing Education (TOCE)*, *22*(3), 1-40. https://doi.org/10.1145/3513140

Ramesh, D., & Sanampudi, S. K. (2022). An automated essay scoring system: a systematic literature review. *Artificial Intelligence Review*, *55*(3), 2495-2527. https://doi.org/10.1007/s10462-021-10068-2

Pérez-Jorge, D., & Martínez-Murciano, M. C. (2022). Gamification with Scratch or App Inventor in Higher education: A systematic review. *Future Internet*, *14*(12), 374.

Martínez-Murciano, M. C., & Pérez-Jorge, D. (2024). Computational thinking with Scratch or App Inventor in primary education. *Campus Virtuales*, *13*(2), 107-122. https://doi.org/10.54988/cv.2024.2.1364

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Exploring How Programming Supports a Student's Spatial Reasoning and Understanding of Quadratic Growth

Xuan SU[1*], Biyao LIANG[2]
[1,2]The University of Hong Kong, Hong Kong SAR
xuansu24@hku.hk, biyao@hku.hk

Abstract: *In the context of the rapidly evolving artificial intelligence era, educational researchers are increasingly exploring the integration of programming tools into mathematics classrooms. However, most studies usually focus on the impact of programming on mathematics learning outcomes and attitudes, and only a few studies focus on how programming affects students' mathematical reasoning processes. This study aims to fill this gap by investigating how engagement in a programming environment influences a student's mathematical reasoning processes, particularly in the context of understanding the second difference of quadratic growth. One seventh-grade student completed the task in two semi-structured interviews, one in a paper-based setting and the other using Scratch. This study used open coding to analyze the student's reasoning process. The findings suggest that, for the student in this study, the programming environment appeared to help restructure and optimize his spatial reasoning, enabling him to explicitly identify the source of the second difference. This study demonstrates the potential of programming as a technological tool to facilitate students' mathematical reasoning and informs how teachers can design function-related tasks in programming environments.*

Keywords: Programming, Mathematical Reasoning, Spatial Reasoning, Quadratic Growth, Second Difference

1. Background

In response to the growing emphasis on Science, Technology, Engineering and Mathematics (STEM) education in the 21st century, many studies have advocated the integration of programming into mathematics curricula (Wang et al., 2022). Unlike the static process in paper-and-pencil environments, programming's interactivity and immediate feedback help students develop logical reasoning and problem-solving through creating, debugging, and modifying code (Kaufmann & Stenseth, 2021; Ng & Cui, 2021). In addition, programming has the potential to support students in communicating through multiple representations, such as words, symbols, and diagrams, making abstract concepts more concrete (DeJarnette, 2019).

Despite these advantages, research has largely focused on the impact of programming on students' learning outcomes and attitudes (Wang et al., 2022), with relatively few studies investigating how programming environments influence students' mathematical reasoning processes. On the other hand, students may find it difficult to express their reasoning in a paper-and-pencil environment, making it difficult for teachers to observe and support their reasoning (Herbert, 2021). This limitation highlights the necessity of exploring other alternative ways to understand students' reasoning processes. Programming environments offer such an opportunity to externalize aspects of students' reasoning by translating their mathematical understanding into explicit, sequential commands (Olteanu, 2022). The tangible artefact of completed code can provide insight into how students organize and interpret mathematical relationships. Therefore, we designed a mathematical task to explore how a student's reasoning processes unfold in a programming environment. The guiding research question is: *How does engagement in a programming environment influence a student's mathematical reasoning processes, particularly in understanding the second difference of quadratic growth?*

2. Methodology

2.1. Participant and Research Design

The participant in this study was Wang (pseudonym), a Chinese seventh-grade student who had not yet learned function at school but was familiar with Scratch. Two one-on-one semi-structured task-based interviews were conducted after school, each lasting 1.5 hours. In each interview, Wang was asked to complete specified mathematical tasks, and the researcher guided his thinking with questions like, "Why do you think so?" and "Why do you do this?"

2.2. Task Design

In order to explore students' mathematical reasoning processes, a task was devised in which the first and second differences of a quadratic function were placed in a growing square pattern. Given that mathematical reasoning may play a particularly important role in connecting numerical and spatial representations (Morsanyi et al., 2018), quadratic growth includes both representations to provide a suitable context for this study.

The task consisted of two activities, one in a paper-and-pencil environment and the other in a programming environment, corresponding to the two interviews. To clarify the relevant mathematical concepts, we used the quadratic function $y = x^2$, where x represents the side length of a square and y represents the area, with a growing square as the figural pattern (*Figure 1*). For this case, assuming the side length increases by 2 each time, the area would be 1, 9, 25, 49... The first difference of the area would be 1, 8, 16, 24..., and the second difference remains constant at 8. This framework guided Wang's exploration of the second difference.

In the first activity, Wang drew squares with increasing side lengths on gridded paper and analyzed changes in side lengths, first differences (area growth), and second differences (change of growth rate). In the second activity, Wang recreated one selected growth pattern using Scratch's cloning feature. The initial code included a parent square of side length 30 as the unit (*Figure 2*). Wang programmed the animation, explained the logic of his code, and identified the second difference within the animation.

2.3. Data Collection and Analysis

Data collection included audio and video recordings, Scratch code, and written work. Wang's reasoning processes were analyzed through open coding using MAXQDA, with memos tracking cognitive shifts. This preliminary analysis serves as the foundation for refining codes and identifying themes to develop Wang's cognitive model in future work.

3. Results

Wang explored several different square growth patterns by drawing a series of shapes on paper in the first activity. Among them, we chose the concentric square growth (*Figure 1*) for detailed reporting because Wang chose this growth pattern to replicate in the programming activity. This section describes Wang's reasoning in both activities.

3.1. Wang's Reasoning in Activity 1

At the beginning of the first activity, Wang relied on numerical reasoning to analyze changes in the area of a square. He set the sides of the concentric square growth to increase of 2 starting from 1, and systematically calculated the areas of four squares with sides 1, 3, 5, and 7 to be 1, 9, 25, and 49, respectively, and then he further calculated the increments of the areas to be 8, 16, and 24. Finally, he did the difference to get that the second difference was always 8 (*Figure 3a*).

When the researcher further guided Wang to explain where the number 8 came from in the figural pattern, Wang switched to a spatial reasoning. He divided the square into 1×1 cells according to the size of the parent, and quickly painted two 1×4 black rectangles on the left and right sides of the figural pattern (*Figure 3b*). Then he gave the following explanation, "Here are these three little squares (points to the top three squares in the red circle), if you just slide them up a bit, they'll cover this part (points to the top three squares in the purple circle). Then these six squares (counts the remaining six squares in the red circle), you can take them and arrange them side by side, and slide them down here (points to the row of squares under the purple circle). After that, what's left is just this 2×4 block, right?"

These observations suggest that Wang attempted to explain the source of the second difference through spatial transformations. However, he mistakenly included the original 1×1 blue square in the first difference (red circle), leading to an overestimation of the red circle's area. Specifically, the red circle should have contained five squares, but Wang counted it as six. This indicates that he had not fully understood that the second difference corresponds to the area difference between successive layers. In addition, while he reasoned that the remaining six squares from the red circle could be rearranged and shifted to fill the bottom of the purple circle, there were in fact only five squares available in that position. Therefore, although Wang exhibited some spatial reasoning abilities, his reasoning largely relied on intuitive visual matching rather than on constructing a precise connection between geometric configurations and changes in area.

3.2. Wang's Reasoning in Activity 2

In the programming environment, Wang initially assigned each clone a specific sequence of moves to build the growing square. Starting from the last square in the previous layer, each new square moved left, up, right, down and then left again to complete a surrounding frame. For example, after creating the initial 1×1 square, it moved 1 unit to the left, 1 unit up, then 2 units to the right, down and left to form a 3×3 square (*Figure 3c*).

After constructing squares with side lengths of 1, 3, 5, and 7 in turn, Wang noticed that his code was becoming increasingly lengthy. He thought there may be a pattern that allows the code to be shorter. Wang carefully examined the values in the code and the corresponding animations, and reasoned by considering himself as the parent square. He repeated his reasoning using the example of a square with side length 3, "So, the parent square needs to move one step to the left to get into the next layer, and then one step up. Because the whole side length is 3, and I'm starting from the center, I need to go 'side length minus 2' steps up. After that, it goes two steps to the right. Since the previous square already takes up one step, it's actually 'side length minus 1' steps. It's the same for going down and left, each time it's 'side length minus 1' steps." This reasoning suggests that Wang moved beyond intuitive visual matching toward rule-based spatial structuring.

Building on this discovery, Wang created a variable named "side length" to dynamically represent the movement steps in his code. He further integrated loop structures to simplify the repeated movement sequences. Finally, he added a conditional statement that terminated the construction process once the figure reached the target side length (*Figure 4*).

After completing the programming task, Wang was once again asked to identify where the second difference "8" appeared in the growing square structure. This time, Wang systematically analyzed how each circle contributed to the increase in area, aligning his reasoning with the movement structure coded in his program. Wang explained the origin of the first "8", "First, when going up, the first circle moves up by 1 step, and the second circle moves up by 3 steps. So that's 2 more squares up. Then to the right, the first one moves 2 steps, and the second one moves 4 steps, so that's another 2 squares. Same for going down and to the left—first 2 steps, then 4 steps, so again 2 more squares. So overall, that's 8 squares added." Thus, Wang reconstructed his understanding of the second difference by recognizing that the total increase of 8 was evenly distributed across the four directions, with each direction contributing 2 additional squares.

4. Discussion and Conclusion

This study explored how engagement in a programming environment influenced a student's mathematical reasoning processes, particularly in understanding the second difference within a quadratic growth context. The findings reveal a clear transformation in Wang's reasoning trajectory from intuition-based spatial reasoning in a paper-and-pencil environment to more structured and rule-based spatial reasoning through programming.

Olteanu (2022) noted that programming activities enable students to externalize mathematical thinking through dynamic constructions. Wang's case confirms this, as his mathematical reasoning process became visible and readily analyzable through the structure of his code. Wang's case further illustrates how the programming environment fundamentally reshaped his spatial reasoning. This shift is closely related to programming logic, which emphasizes moving from the concrete to the abstract (DeJarnette, 2019). For example, the unique looping feature of the programming environment prompted Wang to actively look for patterns to simplify his construction when faced with increasingly long code, thereby facilitating pattern recognition and structural reasoning. Similarly, the ability to create variables provided a mechanism for associating numerical quantities with spatial transformations. For example, Wang created a "side length" variable to relate the side length to the number of movement steps.

Furthermore, Hernández-Zavaleta et al. (2023) emphasized that programming environments can enable students to manipulate sprites' movements from a first-person perspective, integrating dynamic sense-making with geometric concepts such as angles and distances. This is supported by the case of Wang in this study. The programming environment encouraged Wang to assume that he was the parent square, which prompted him to focus on the dynamic direction of movement. This shift in perspective enabled him to later decompose the second difference into four directional increments.

This study provides practical insights for educators on using tasks that include both numerical and spatial elements when designing programming-based mathematical tasks to promote deeper mathematical reasoning among students. However, this study is based on a single student's engagement with a specific task, limiting the generalizability of the findings. Future research involving a wider range of students, different types of programming tasks, and longer intervention periods will help to elucidate how programming environments systematically influence the development of mathematical reasoning.

References

DeJarnette, A. F. (2019). Students' challenges with symbols and diagrams when using a programming environment in mathematics. *Digital Experiences in Mathematics Education*, *5*, 36–58. https://doi.org/10.1007/S40751-018-0044-5

Herbert, S. (2021). Overcoming Challenges in Assessing Mathematical Reasoning. *Australian Journal of Teacher Education*, *46*(8), 17-30. https://doi.org/10.14221/ajte.2021v46n8.2

Hernández-Zavaleta, J. E., Brady, C., Becker, S., & Clark, D. B. (2023). Vygotskian hybridizing of motion and mapping: Learning about geometric transformations in block-based programming environments. *Mathematical Thinking and Learning*, 1-35. https://doi-org/10.1080/10986065.2023.2191074

Kaufmann, O. T., & Stenseth, B. (2021). Programming in mathematics education. *International Journal of Mathematical Education in Science and Technology*, *52*(7), 1029–1048. https://doi.org/10.1080/0020739X.2020.1736349

Morsanyi, K., Prado, J., & Richland, L. E. (2018). Editorial: The role of reasoning in mathematical thinking, *Thinking & Reasoning*, *24*(2), 129-137, https://doi.org/10.1080/13546783.2018.1435425

Ng, O. L., & Cui, Z. (2021). Examining primary students' mathematical problem-solving in a programming context: Towards computationally enhanced mathematics education. *ZDM - Mathematics Education*, *53*(4), 847–860. https://doi.org/10.1007/s11858-020-01200-7

Olteanu, C. (2022). Programming, mathematical reasoning and sense-making. *International Journal of Mathematical Education in Science and Technology*, *53*(8), 2046-2064. https://doi-org/10.1080/0020739X.2020.1858199

Wang, C., Shen, J., & Chao, J. (2022). Integrating Computational Thinking in STEM Education: A Literature Review. *International Journal of Science and Mathematics Education*, *20*(8), 1949–1972. https://doi.org/10.1007/S10763-021-10227-5/METRICS

Appendix

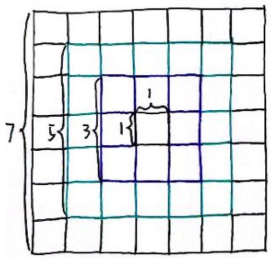


Figure 1. Figural pattern of the growing square.

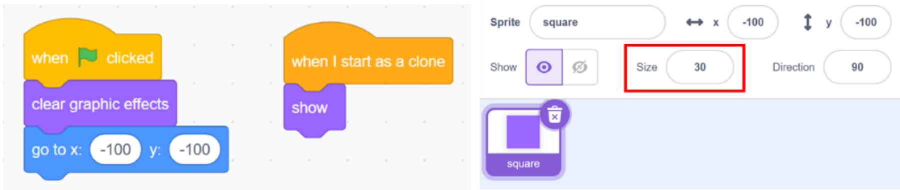


Figure 2. Initial code and parent square provided to the student.

(a) (b) (c)

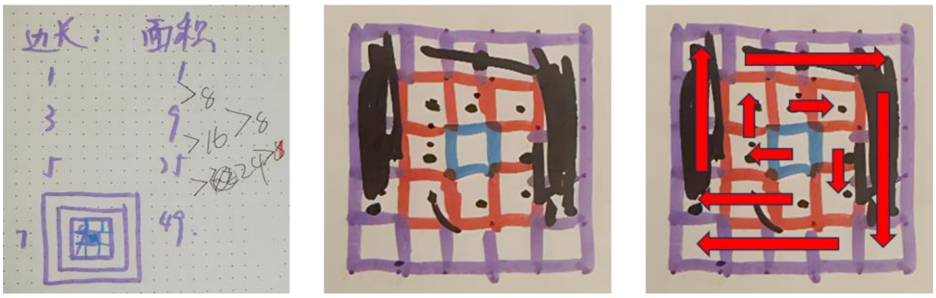


Figure 3. Wang's paper-and-pencil reasoning about growing square patterns.

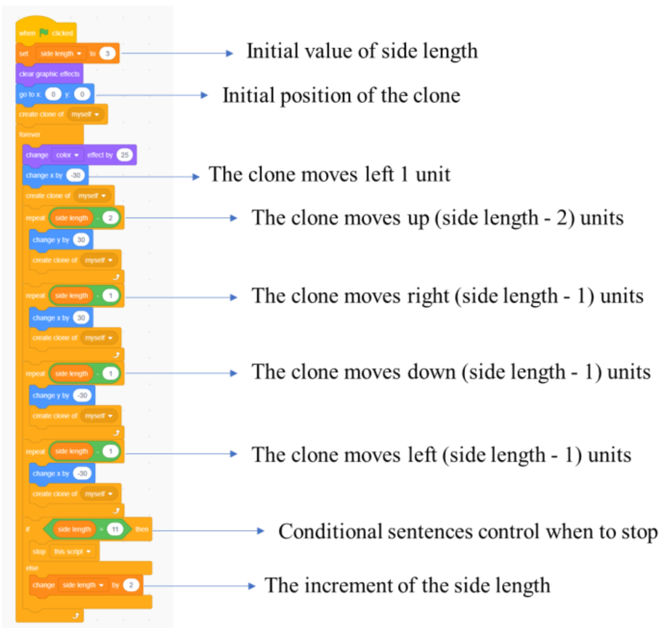


Figure 4. Wang's final code to show the animation of the concentric square growth.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Generative AI and Four-Learning Teaching Applications: Information Technology Tower of Hanoi Algorithm

Tsun-Sheng FAN[1], Ting-Chia HSU[2], Yi-Wen LIAO[3*]
[1]Saint Peter Senior High School, Taiwan
[2]National Taiwan Normal University, Taiwan
[3*]National Kaohsiung University of Science and Technology, Taiwan
[1]p041234@sphs.hc.edu.tw, [2]ckhsu@ntnu.edu.tw, [3*]pinkwen923@gmail.com

Abstract: *This study aims to explore how generative AI tools can be integrated into the teaching of the Tower of Hanoi algorithm by incorporating the "Four-Pillar" educational approach: self-regulated learning, collaborative learning, mutual learning, and guided learning. A curriculum package was designed on the ADL (Adaptive Learning) digital learning platform, including video instruction, gamified exercises, group discussions, AI-assisted learning, and programming tasks, guiding students to derive the Tower of Hanoi algorithm. Through the digital platform, students not only learn the rules and solutions through gamification but also gain an understanding of the underlying algorithm and apply it in practice. The results show that this model effectively enhances learning motivation and computational thinking, promotes the deep integration of information technology into teaching, and analyzes the supportive role of generative AI tools (such as e 度) in self-regulated learning and problem-solving.*

Keywords: Tower of Hanoi, Algorithm, Computational Thinking, Four-Pillar Learning Model, Adaptative Learning (ADL)

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

生成式 AI 與四學教學應用:資訊科技河內塔演算法

范村生 [1]，許庭嘉 [2]，廖奕雯 [3*]
[1] 新竹市天主教磐石高級中學，臺灣
[2] 臺灣師範大學，臺灣
[3] 高雄科技大學，臺灣
[1]p041234@sphs.hc.edu.tw，[2]ckhsu@ntnu.edu.tw，[3] pinkwen923@gmail.com

【摘要】 本研究旨在探討如何將生成式 AI 工具融入河內塔演算法教學，結合自學、共學、互學、導學的「四學」教育理念，設計一套因材網數位學習平台課程包。課程內容包括影片學習、遊戲實作、分組討論、AI 輔助學習與程式設計，引導學生推導河內塔演算法。透過數位平台，學生不僅以遊戲化方式學習規則與解法，亦能理解演算法原理並應用於實作。研究結果顯示，此模式能有效提升學習動機與運算思維，促進資訊科技深度融入教學，並分析生成式 AI 工具（如 e 度）在自主學習與解題中的輔助角色。

【關鍵字】 河內塔；演算法；運算思維；因材網；生成式 AI_e 度

1. 前言

隨著資訊科技快速發展，如何有效融入教學、提升學習成效成為教育挑戰。河內塔為經典遞迴演算法問題，能訓練學生邏輯與運算思維 (Waskita & Rifai, 2023)。臺灣教育部建置的因材網結合課綱、知識地圖與個別化學習，透過自學、共學、互學、導學（四學）模式提升學生表現 (Kuo, Chang & Lee, 2023)。2024 年開放的 e 度功能，結合 ChatGPT 與蘇格拉底提問法，作為 AI 學習夥伴輔助學生線上學習 (Wu et al., 2025)。本研究以四學理念為基礎，結合因材網與生成式 AI 工具，設計河內塔演算法教學活動，探討此模式對學生學習動機、運算思維與資訊科技應用的成效。

2. 文獻探討

2.1. 運算思維

運算思維是一種應用電腦科學概念解決問題的思維模式，包括問題拆解、模式識別、抽象化與演算法設計 (Wing, 2006)。在河內塔教學中，學生透過蘇格拉底提問法引導，進行演算法推導、問題分析與程式實作，藉由錯誤觀察與反覆驗證培養運算思維 (Waskita & Rifai, 2023)。河內塔由 Lucas 於 1883 年提出，具遞迴特性，是學習演算法與運算思維的理想教材。近期研究指出，結合蘇格拉底提問法的生成式 AI 學習夥伴，能有效促進學生深度學習與高階思維發展，展現未來科技教育的潛力 (Wu et al., 2025)。

2.2. 適性學習與因材網

適性學習透過提供符合學習者需求的客製化資源,促進個別化學習 (Chen et al., 2024; Kuo, Hsieh et al., 2024)。因材網為一數位學習平台，內建生成式 AI 聊天機器人「e 度」，支援知識結構圖與學習建議，幫助學生依據自身步調學習 (陳志鴻, 2024)。平台結合自主學習、共學、互學、導學（四學）理念，強調主動探索、小組合作、同儕觀摩與教師引導 (Kuo, Chang & Lee, 2023)，達成適性化學習效果。研究指出，e 度結合蘇格拉底提問法能提供即時回饋，促進學生深度學習與學習動機 (Ho, 2022; Hung & Huang, 2022)。本研究以 e 度與因材網為基礎，探

討生成式 AI 在適性學習與自主學習中的應用，協助學生透過平台進行個別化、互動式學習，教師亦能利用 AI 優化教學設計與指導。

3. 研究方法

3.1. 研究對象

本研究以高中十年級普通科學生為研究對象，男生 25 人，女生 20 人，年齡為 15-16 歲，選修校定必修「風城走讀」課程中「演算法-河內塔」單元。

3.2. 教學設計

本研究以因材網平台為基礎，設計以河內塔為主題的課程包，結合線上遊戲、教學影片、討論區與 e 度 AI 學習夥伴，提供即時學習輔助。課程設計包含：(1) 課前觀看影片並透過 e 度初步了解；(2) 自主操作線上河內塔遊戲並分享心得；(3) 組內討論推導演算法並用 e 度驗證；(4) 組間展示與觀摩學習；(5) 於 Colab 實作 Python 程式並驗證；(6) 討論區分享反思；(7) 教師引導推導演算法並透過平台檢視學習狀況，針對學生問題進行個別化指導。此教學模式結合多元資源與 AI 輔助，強調自主、合作與互動學習，促進學生運算思維與問題解決能力。

3.3. 資料收集

本研究主要透過以下方式收集資料：(1) 學生在因材網討論區的發言記錄。(2) 學生操作河內塔遊戲的數據。(3) 學生與 e 度 AI 學習夥伴的互動紀錄。(4) 學生產出的 Python 程式碼。(5) 學生學習心得與反思。

3.4. 教學方式

課程教學方式，學生透過觀看教學影片，課堂中老師進行講解，學生實際操作河內塔解題，最後進行課堂討論，如表 1. 所示。

表1. 教學方式、設備、資源

	第一週	第二週	第三週
教學方式	觀看影片	老師講解	實際操作、課堂討論
教師授課資源	簡報	因材網課程包	
觀看影片/課堂討論設備	電腦 PC	平板	
學生學習任務	因材網課程包	討論區	上台報告
使用平台	因材網數位學習平台		
教學總時間	每週 2 節，共計三週		

3.5. 教學策略

本教學活動採高引導與高協作模式，結合因材網共享協作平台、討論區及數位說故事進行。教學流程依四學模式設計：(1) 自學：觀看河內塔相關影片（如《猩球崛起》片段）、操作線上遊戲記錄步數；(2) 組內共學/導學：小組於討論區回答問題、運用 e 度 AI 與運算思維推導演算法並驗證；(3) 組間互學/導學：各組展示演算法流程，分享邏輯與推理過程；(4) 教師導學：引導學生透過 Colab 平台將推導演算法轉化為 Python 程式，體驗演算法的實作。教學中學生與 e 度互動求證、教師進行即時引導，促進學生運算思維與合作學習。

4. 研究結果

研究顯示，透過遊戲化學習，學生對河內塔演算法有更深入理解，能運用分解、模式識別與抽象化推導演算法並轉化為程式實作。在四學應用中，學生透過影片與 e 度 AI 自學，於小組中討論與驗證想法，組間展示促進知識交流，教師則透過討論區觀察與回饋，深化學習。e 度 AI 提供即時回饋與個人化建議，提升學習效率與自主性，並協助教師掌握學習狀況、調整教學策略。因材網平台提供豐富教學資源與互動工具，有效提升學生學習動機與參與度。學

生透過討論區分享想法、提問，促進師生互動。平台的報表功能(如圖 1 所示) 協助教師即時掌握學習進度，進行個別化指導，並於課程結束後收集學生心得與反思，如圖 2 所示。

圖1 AI 學習夥伴報表 (https://adl.edu.tw)

圖2 同學心得與反思 (https://adl.edu.tw)

4.1. 自主學習及反思學習單分析結果

本研究結合資訊科技、四學理念與生成式 AI（e 度），設計並實踐河內塔演算法數位學習課程。結果顯示，此課程能有效提升學生自主學習、學習成效與運算思維。學生透過影片、遊戲實作、AI 輔助與分組討論，增進對演算法的理解與程式實作能力。學生多以實作、討論與 AI 互動作為學習策略，分組討論有助觀點交流、邏輯推理與學習動機。學生反思指出需加強自律與時間管理，建議增加實作時間、提供更詳細指引與強化 AI 個人化功能。AI 在教學中扮演學習助理角色，提供即時回饋、可視化解釋與程式錯誤提示，提升學習效率。

5. 結論與建議

本研究成功將資訊科技、四學理念（自學、共學、互學、導學）與生成式 AI 學習夥伴（e 度）融入河內塔演算法教學，設計出一套高效教學模式。研究結果顯示，此模式能顯著提升學生的學習動機、運算思維與程式設計能力，並促進學習成效。課程透過數位學習平台、遊戲化學習、分組討論與 AI 輔助學習結合，增進自主學習、協作與問題解決能力。e 度在教學中提供即時回饋與個人化建議，提升學習效率。未來此模式可應用至其他資訊科技課程，教師亦應持續精進資訊科技素養以強化教學。建議未來研究深入分析學習數據，探索不同學習背景對學習成效的影響，並推廣此模式至多元學科與教育情境，持續優化數位學習平台與生成式 AI 應用，以強化個別化學習與教育成效。

參考文獻

陳志鴻 (2024)。人工智慧輔助多媒體教學設計與實踐。 **臺灣教育評論月刊**, **13**(11), 50-54。

Chen, M. S., Hsu, T. C., & Tu, Y. F. (2024). The use of learning analytics and educational data mining to analyze teachers’ teaching in online environments: a systematic literature review. *Interactive Learning Environments*, 1-16.

Ho, S. C. (2022). Curriculum practice of science and technology assisting primary school children's independent learning. *School Administrators*, *140*, 103-121.

Hung, S. H., & Huang, J. W. (2022). A study on the use intention of elementary school students in adaptive learning platform based on technology acceptance model. *School Administrators, 139*, 114-133.

Kuo, B. C., Chang, F. T. Y., & Lee, Y. L. (2023). Trends and issues of digital learning in Taiwan. *Trends and Issues of Promoting Digital Learning in High-Digital-Competitiveness Countries: Country Reports and International Comparison*, 349-392. https://files.eric.ed.gov/fulltext/ED636595.pdf#page=357

Kuo, B. C., Hsieh, F. L., Liao, C. H., & Wu, P. C. (2024). A dialogue-based intelligent tutoring system of reading inference for elementary school students. *Psychological Testing*, *71*(3), 331-361. https://www.airitilibrary.com/Article/Detail/16094905-N202409270006-00002

Waskita, A., & Rifai, H. (2023). Needs analysis of design for the integrated project-based learning module at Edupark physics science andt technology center Sawahlunto. *Physics Learning and Education*, *1*(4), 217-224. https://doi.org/10.24036/ple.v1i4.94

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35. https://doi.org/10.1145/1118178.111821

Wu, T. T., Lee, H. Y., Chen, P. H., Lin, C. J., & Huang, Y. M. (2025). Integrating peer assessment cycle into ChatGPT for STEM education: A randomised controlled trial on knowledge, skills, and attitudes enhancement. *Journal of Computer Assisted Learning*, *41*(1), e13085. https://doi.org/10.1111/jcal.13085

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Integrating Computational Thinking in Indian K-12 Education Under NEP 2020

Sabitha VINOD
Meghe Group of Schools, Nagpur, Maharashtra, India
sabita.rcict@mgsnagpur.org

Abstract: *Computational Thinking (CT) has emerged as a crucial 21st-century skill, considered fundamental for all students. Nations worldwide have responded by integrating CT and coding into school curricula, with over 30 countries mandating computing education from early grades. India's National Education Policy 2020 (NEP 2020) similarly emphasizes CT and coding as essential, introducing these concepts from the middle school level. This policy push has catalyzed efforts to weave CT into K-12 education, but significant challenges persist. A lack of a structured CT curriculum framework, limited classroom time and computing resources, and gaps in teacher training hinder effective implementation. This paper reviews the current state of CT education in India in light of NEP 2020, identifies key challenges in integrating CT into primary and middle school (Grades 1–8), and offers recommendations for establishing a comprehensive CT framework. The recommendations focus on curriculum development, cross-curricular integration with mathematics and science, teacher professional development, and inclusion of emerging topics like artificial intelligence. Despite present challenges, India's proactive policies and growing grassroots initiatives provide a positive outlook for embedding CT across the education system. Establishing a clear framework and support system will enable Indian schools to empower every student with CT skills for the future.*

Keywords: Computational Thinking; NEP 2020; K-12 CT Curriculum; CT Pedagogy; Teacher Training, Capacity Development

1. Introduction

Computational Thinking (CT) involves skills like decomposition algorithmic thinking and abstraction. It equips students to approach problems logically, much like computer scientists, even without direct coding skills. Globally, CT has been increasingly recognized as a vital skill, with over 30 countries formally integrating it into their school curricula and India's National Education Policy (NEP) 2020 has also adopted this global trend by introducing CT and coding from grades 6 through 8. This paper reviews the present state of CT education in Indian primary and middle schools, examines the challenges educators face in implementation, and provides practical recommendations to effectively embed CT into the curriculum, building on the momentum set by NEP 2020.

2. NEP 2020's Emphasis

The introduction of the National Education Policy (NEP) 2020 was pivotal, formally establishing CT and coding as central elements of Indian education. The policy specifically introduces coding from Grades 6–8, encouraging students to practice algorithmic thinking and logical reasoning regularly. With NEP 2020's directive, educators and curriculum developers have begun embedding fundamentals of coding, robotics, and introductory concepts in emerging technologies like artificial intelligence within the curriculum. The goal goes beyond mere digital literacy, focusing instead on nurturing computational problem-solving abilities across student populations.

2.1. Pre-NEP Initiatives by non-profit agencies (CSpathshala)

Prior to NEP 2020, grassroots initiatives like CSpathshala, launched by ACM India Council in 2016, had already begun promoting CT in schools. CSpathshala introduced a curriculum emphasizing "unplugged" methods, allowing students to learn computing concepts without needing extensive technological infrastructure. By 2019, it reached hundreds of schools nationwide, significantly influencing state education boards like Tamil Nadu, where CT was integrated into mathematics for approximately 10,000 schools. Despite substantial teacher training and local-language resources, these early efforts remained voluntary, leading to inconsistent adoption.

2.2. Post-NEP Adoption Surge

Following NEP 2020, CT gained broader momentum in Indian education. The Central Board of Secondary Education (CBSE) rolled out coding curricula from Grades 6–8, focusing on project-based and visual programming approaches. Concurrently, many private institutions-initiated robotics and coding courses, and educational publishers updated textbooks with new CT activities. However, significant disparities persist—urban schools have swiftly integrated CT, while rural schools continue to struggle with infrastructure limitations and insufficient teacher preparation, highlighting the ongoing need for a standardized, nationwide framework.

2.3. Global Comparison: India, UK & China

The UK Computing Curriculum mandates CT from age 5, supported by CPD for teachers via NCCE. China, since 2017, embedded CT and AI education into its national system with tech partnerships and clear grade-wise outcomes. Compared to India's decentralised and delayed model, both countries show the impact of early integration, state backing, and consistent teacher training. Their models can guide India's CT roadmap, especially for CIE-aligned institutions.

3. Challenges in Implementing CT in India

Despite enthusiasm for CT in principle, Indian schools face several challenges in translating policy into practice. Key issues include:

3.1. Absence of a Structured CT Curriculum (Grades 1–8)

Currently, India lacks a detailed, nationally standardized curriculum for computational thinking (CT) in Grades 1–8. Although NEP 2020 broadly encourages integrating CT into school curricula, there is no clear roadmap specifying what exact skills should be taught at each grade. Consequently, schools and publishers develop their own interpretations, resulting in inconsistent and fragmented teaching approaches.

3.2. Limited Classroom Time for ICT/CT

In many primary and middle schools, the timetable allocated for ICT classes are often just two class per week. These ICT classes traditionally focus on basic computer usage or applications, leaving scant time for CT activities. With such constraints, it is challenging to engage students in meaningful CT practice beyond superficial exposure. Without increasing or reprioritizing time for CT, schools struggle to go deeper than simple demonstrations.

3.3. Shortage of Devices and Infrastructure

A major practical barrier in India is the lack of sufficient computing devices and infrastructure in schools, especially outside of well-funded urban institutions. As of a few years ago, only roughly 27% of schools in India had computers, often outdated, and limited in number. This digital divide makes it difficult to implement CT activities and forces most schools to rely on low-tech or unplugged methods.

3.4. Textbook Content and Proprietary Focus

ICT textbooks in India predominantly focus on teaching students proprietary software skills rather than fundamental computational thinking. Textbooks often emphasize procedural software usage, neglecting deeper problem-solving, logical reasoning, or algorithmic thinking. The integration of free or age-appropriate coding resources is limited, further hindering effective CT development among students (Table 1).

Table1. – Textbook Analysis

Grade Level	Computational Topics Covered	Observations
I - II	Basic literacy, introductory coding	Minimal CT integration
III - V	Productivity software, block coding	Superficial CT coverage
VI - VIII	Web basics, ethics, basic AI concepts	Limited authentic CT application

3.5. Pedagogical and Teacher Training Gaps

A critical challenge in implementing CT is the limited preparedness of teachers. India's teacher education programs (e.g., B.Ed.) traditionally lack computational thinking and computer science training. Consequently, ICT teachers often lack formal CT pedagogical skills, and computer instructors with technical backgrounds may lack familiarity with effective teaching methods for younger learners. Effective implementation thus necessitates robust professional development in CT pedagogy to ensure teachers can confidently deliver the curriculum.

4. Recommendations for a CT Framework

To successfully integrate computational thinking into India’s K-12 education, a holistic framework is needed, supported by curriculum guidelines, teacher development, and infrastructure improvements. Below are key recommendations:

4.1. Develop a National CT Curriculum (Grades 1–8)

A clear national CT curriculum should define grade-specific computational thinking skills, ensuring consistent learning across schools. Inspired by international frameworks like CSTA, the UK Computing Curriculum, and China’s national AI roadmap, it should emphasize logical reasoning, practical problem-solving, and unplugged activities beyond coding alone. The upcoming revisions of India's National Curriculum Framework (NCF) offer a timely opportunity to introduce structured CT standards, ensuring aligned textbooks and coherent skill progression nationwide.

4.2. Curriculum Content on Emerging Topics

The CT curriculum should introduce contemporary computing topics like Artificial Intelligence (AI) and data science at age-appropriate stages. Middle-school students, for instance, could engage in simple, conceptual activities such as classifying images to understand abstraction or exploring basic data analysis to foster logical thinking. This practical, inquiry-based approach can make CT relevant and engaging, aligning with India's emphasis on future-ready education and innovation.

4.3. Integrate CT within Mathematics and Science

The curriculum framework should integrate computational thinking (CT) within mathematics and science classes rather than isolating it as a standalone subject. Activities such as using pattern recognition in mathematics or analyzing data sets in science lessons naturally embed CT skills into everyday learning. This approach ensures consistent practice, maximizes limited ICT class time, and helps students appreciate CT as an integral, practical problem-solving method across subjects. Successful examples, like Tamil Nadu's integration of CT into math education, show this approach is effective and scalable in the Indian context.

4.4. Teacher Training and Capacity Building

Effective CT integration depends heavily on teacher preparation. A comprehensive approach, including both in-service workshops and pre-service teacher training, should be established. In-service workshops should focus on CT pedagogy, classroom techniques, and practical resources, while pre-service (B.Ed.) programs must incorporate essential CT modules. Strengthening teacher competencies will significantly enhance effective CT instruction in classrooms.

4.5. Policy and Administrative Support

Effective CT implementation requires firm backing from education authorities. National and state boards should formally adopt CT competencies and provide clear school guidelines, including regular assessments or project-based tasks to reinforce implementation. Continued investment in infrastructure—such as computer labs and reliable internet—is crucial, and public-private partnerships should support curriculum development, resource provision, and teacher training. Systematic evaluation of pilot initiatives will enable policymakers to scale successful practices uniformly.

With these coordinated efforts—a structured curriculum, integrated teaching methods, empowered teachers, resource availability, and supportive policy—India can effectively embed computational thinking into its education system, laying a strong foundation for future learners.

5. Conclusion

Computational Thinking (CT) is a key 21st-century skill, and NEP 2020 has emphasized its role in school education. While progress has been made, challenges like curriculum gaps, resource constraints, and inadequate teacher training persist. Addressing these through a structured CT framework, cross-curricular integration, teacher development, and infrastructure support—especially with low-tech solutions—can ensure widespread adoption. With collective efforts from policymakers, educators, and stakeholders, India can equip students with essential computational skills for a technology-driven future.

References

Ministry of Education, Government of India. (2020). *National Education Policy 2020*. New Delhi

Berry, M. (2013). *Computing in the national curriculum: A guide for primary teachers*. Computing at School.

China Ministry of Education. (2018). *Action Plan for Artificial Intelligence Innovation in Schools*.

Shah, V. (2019). CSpathshala: Bringing computational thinking to schools. *Communications of the ACM, 62*(11), 30–34. https://doi.org/10.1145/3363187

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. http://doi.acm.org/10.1145/1118178.1118215

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Metacognitive Awareness, Self-Regulation Confirmation, Inventive Self-Efficacy and Continuous Self-Improvement: Differences in School Grade from an Invention Exhibition

Jia Sheng NGAI[1], Jon Chao HONG[2*]
[1] Chang Gung University, Taiwan
[2]National Taiwan Normal University, Taiwan
jason.js.ngai@gmail.com, tcdahong@gmail.com

Abstract: *The International Exhibition for Young Innovators (IEYI) serves as a unique platform for young innovators to showcase their STEM-driven inventions, fostering creativity and collaboration. Celebrating its 20th anniversary in 2024, this study investigates the impact of metacognitive awareness, self-regulation confirmation, inventive self-efficacy and continuous self-improvement on young inventors' development. Through the lens of STEM education, these cognitive factors are crucial for addressing complex problems, driving innovation, and enhancing problem-solving abilities. This study explores how these elements influence the design, iteration, and evaluation of inventions in a competitive setting, providing valuable insights into the role of metacognitive processes in shaping the cognitive and creative abilities of young inventors.*

Keywords: Metacognitive Awareness, Inventive Self-Efficacy, Self-Regulation Confirmation, Continuous Improvement, STEM Education

1. Introduction

Competitions like the International Exhibition for Young Innovators (IEYI) challenge participants to not only demonstrate their technical skills but also apply higher-order thinking processes to define objectives, set goals, and solve complex STEM problems. In these settings, participants must use metacognitive strategies to navigate the design process, analyze results, and refine their solutions. During adolescence, cognitive and psychosocial development accelerates, enhancing the ability to engage in abstract thinking, critical problem-solving, and self-regulation (Barbot & Heuser, 2017). These advancements are particularly impactful in STEM fields, where creativity and innovation are essential to developing new technologies and inventions (Ma et al., 2024). One notable example of STEM in real-world innovation is the development of low-cost 3D-printed prosthetic hands for amputees. This invention applies science by studying human hand anatomy and biomechanics, technology through 3D modeling and printing, engineering in designing a functional and durable structure, and mathematics in calculating joint movements and grip strength (Nibe et al., 2023). In the Taiwan Youth Invention Exhibition, students from different academic levels—elementary, junior high, and high school—collaborate to design and build STEM-focused devices, using metacognitive strategies to meet the competition's rigorous criteria.

2. Methodology

The study employed a purposive sampling method, resulting in a total of 192 responses, of which 185 were valid, yielding a valid response rate of 96.35%. Among the participants, 82 were male (44.3%), and 103 were female (55.7%). The majority of participants (123; 66.5%) were aged 15, while 62 participants (33.5%) were aged 16.

To analyze differences among grade (elementary, junior high, and high school) across Metacognitive Awareness, Inventive Self-Efficacy, Self-Regulation Confirmation, and Continuous Improvement Intention, a one-way ANOVA was conducted. Post hoc comparisons were performed using Scheffe's test to identify significant group differences.

3. Result

A one-way ANOVA was conducted to examine the differences among different grade (elementary, junior high, and high school) in terms of Metacognitive Awareness, Inventive Self-Efficacy, Self-Regulation Confirmation, and Continuous Improvement Intention. The results are summarized in Table 1.

Table 1: School grade differences across constructs.

Construct	School Grade	N	M	SD	F	P	Post hoc test
Metacognitive Awareness	Elementary	43	4.27	0.58	5.06**	0.007	3>2
	Junior High	53	4.08	0.70			
	Senior High	82	4.39	0.44			
Self-Regulation Confirmation	Elementary	43	4.28	0.48	6.32**	0.002	3>2
	Junior High	53	4.05	0.79			
	Senior High	82	4.41	0.43			
Inventive self-efficacy	Elementary	43	4.19	0.56	5.15**	0.007	3>2
	Junior High	53	4.00	0.71			
	Senior High	82	4.34	0.54			
Continuous self-improvement	Elementary	43	4.26	0.51	5.58**	0.004	3>2
	Junior High	53	4.19	0.71			
	Senior High	82	4.49	0.47			

Metacognitive Awareness showed a significant difference among groups (F = 5.06**, p = 0.007). Post hoc analysis using Scheffe's test indicated that high school students (M = 4.39, SD = 0.44) had significantly higher metacognitive awareness than junior high school students (M = 4.08, SD = 0.70).Self-Regulation (Confirmation) also demonstrated a significant difference (F = 6.32**, p = 0.002), with high school students (M = 4.41, SD = 0.43) scoring significantly higher than junior high school students (M = 4.05, SD = 0.79).Similarly, Inventive Self-Efficacy differed significantly across groups (F = 5.15**, p = 0.007). High school students (M = 4.34, SD = 0.54) exhibited greater self-efficacy compared to junior high school students (M = 4.00, SD = 0.71). Finally, Continuous Improvement Intention was significantly different among groups (F = 5.58**, p = 0.004). High school students (M = 4.49, SD = 0.47) showed a higher intention for continuous improvement than junior high school students (M = 4.19, SD = 0.71). Overall, the results indicate a trend where high school students consistently report higher scores across all variables compared to junior high school students. No significant differences were found between elementary and high school students.

4. Discussion

The findings of this study highlight significant differences in Metacognitive Awareness, Inventive Self-Efficacy, Self-Regulation Confirmation, and Continuous Improvement Intention across different grade, emphasizing the developmental impact of STEM education on students' cognitive and inventive abilities. Across all four measured constructs, high school students consistently outperformed junior high school students, suggesting that as students' progress in their academic journey, their ability to self-regulate learning, develop confidence in problem-solving, and engage in iterative improvement processes strengthens. This aligns with prior research indicating that advanced exposure to STEM-related activities enhances students' higher-order thinking skills and self-efficacy (Han et al., 2021). The significant disparity between junior high and high school students underscores a potential developmental bottleneck in STEM education during the junior high years. At this stage, students may encounter challenges in cultivating the confidence and problem-solving skills essential for innovation (Li, 2023). This finding highlights the necessity for targeted interventions in STEM curricula at the junior high level, including project-based learning, hands-on experimentation, and AI-integrated STEM activities. Such pedagogical strategies could enhance students' cognitive engagement and better prepare them for the advanced analytical and inventive demands of higher education. Moreover, the results emphasize the critical role of fostering Continuous Improvement Intention, a fundamental attribute in STEM disciplines where progress is driven by iterative problem-solving and refinement (Ayala et al., 2021). The significantly higher scores observed among high school students suggest that increased exposure to structured STEM experiences—such as science fairs, robotics competitions, and coding challenges—contributes to the development of a mindset centered on lifelong learning and self-directed improvement. These experiences may provide students with opportunities to refine their metacognitive awareness and inventive self-efficacy through repeated cycles of hypothesis testing, feedback incorporation, and innovation.

The findings of this study reinforce the importance of a robust and well-structured STEM education framework in fostering students' cognitive and inventive capabilities. Given the observed disparities across academic levels, future research should investigate the efficacy of specific STEM-based interventions aimed at strengthening metacognitive awareness, inventive self-efficacy, and Self-Regulation Confirmation at earlier stages of education. By implementing evidence-based curricular enhancements, educators and policymakers can work toward a more equitable and progressive development of innovation competencies across different educational levels (Oecd, 2023).5. Main Text and Quotations

6. Conclusions

This study examined differences in Metacognitive Awareness, Inventive Self-Efficacy, Self-Regulation Confirmation, and Continuous Improvement Intention across different academic levels. The results revealed that high school students consistently outperformed junior high school students in all measured constructs, highlighting the role of STEM education in fostering cognitive and inventive skills. These findings suggest that as students advance in their education, their ability to self-regulate learning, develop confidence in problem-solving, and engage in iterative improvement processes improves significantly.

The observed gap between junior high and high school students indicates a critical need to enhance STEM education at the junior high level. Integrating more hands-on learning experiences, AI-assisted problem-solving activities, and interdisciplinary STEM projects could help students develop stronger metacognitive and inventive skills earlier in their academic journey.

References

Ayala, M. J., Carter, J. K., Fachon, A. S., Flaxman, S. M., Gil, M. A., Kenny, H. V., Laubach, Z. M., Madden, S. A., McDermott, M. T., Medina-García, A., Safran, R. J., Scherner, E., Schield, D. R., Vasquez-Rey, S., & Volckens, J. (2021). Belonging in STEM: an interactive, iterative approach to create and maintain a diverse learning community. *Trends in Ecology & Evolution*, *36*(11), 964–967. https://doi.org/10.1016/j.tree.2021.08.004

Barbot, B., & Heuser, B. (2017). Creativity and identity formation in adolescence: A developmental perspective. In *The creative self* (pp. 87–98). Academic Press.

Han, J., Kelley, T., & Knowles, J. G. (2021). Factors influencing student STEM learning: Self-Efficacy and outcome expectancy, 21st century skills, and career awareness. *Journal for STEM Education Research*, *4*(2), 117–137. https://doi.org/10.1007/s41979-021-00053-3

Li, N. (2023). Research on the cultivation of college Students' innovation Ability. *Advances in Educational Technology and Psychology*, *7*(16). https://doi.org/10.23977/aetp.2023.071612

Ma, Y., Zhang, H., & Wang, M. (2024). The effect of parental support on student self-rated and task-based creativity: The mediating role of creative interest and self-efficacy. *Thinking Skills and Creativity*, *52*, 101512. https://doi.org/10.1016/j.tsc.2024.101512

Nibe, N. et al. (2023). Design and development of low-cost myoelectric prosthetic arm for Upper Limb Amputees, *2023 4th International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, pp. 1–6. https://doi.org/10.1109/iccakm58659.2023.10449544

Oecd. (2023). Education and innovation for the digital and green transitions: How higher education can support effective curricula in schools. In *Education Policy Pointers*. https://doi.org/10.1787/3dedf4cb-en

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Research on Design-Based-Augmented Reality Learning for Facilitating Students' Behaviors in Computational Thinking

Shucheng LUO [1*], Xiao-Fan LIN[2]
[1,2]South China Normal University, China
919118249@qq.com, linxiaofan@m.scnu.edu.cn

Abstract: *Computational thinking (CT) is a necessary skill for the 21st century. However, with the lack of effective pedagogy and guidance from visualization tools, students may be challenged to solve complex CT problems in the practices. Accordingly, this study designed a design-based-augmented reality learning approach to mitigate the challenges of CT development. A lag sequential analysis was conducted by recruiting 98 elementary school students to examine the effects of the approach on students' behaviors in developing CT. Results indicated that the proposed approach improved the key behaviors that facilitate students' CT practices, for example, redesigning algorithms, evaluating solutions, decomposing problems and transferring learning. This study provides insights for teachers to update their teaching concepts and approaches to integrate CT into other disciplines of instruction.*

Keywords: computational thinking, augmented reality, behaviors, algorithm, decomposing

1. Introduction

Computational thinking (CT) is a thinking activity that involves problem solving using methods from the field of computer science (Yang et al., 2025), and has been recognized as one of the critical skills necessary for the 21st century. The Compulsory Education Information Technology Curriculum Standards (2022 edition) emphasize explicitly the importance of CT for students' development by including it as one of the core literacy objectives of the discipline. Existing studies have shown that CT practices facilitate the development of students' general skills, such as problem solving, critical thinking, and creative thinking (Wong & Cheung, 2020). However, new CT learners have challenges in generalizing computational concepts and computational practices to solve daily issues. For example, they may unintentionally misinterpret certain concepts and problem situations during the learning and programming process (Kandemir et al., 2021). This study aimed to explore the effects of a design-based-augmented reality (AR) learning approach on students' behavioral patterns in developing CT to facilitate students' CT cognitive processes and practices.

2. Research Design

In this study, a design-based-AR learning strategy was constructed, including five phases: (1) AR creating context; (2) AR finding keys; (3) AR facilitating exploration; (4) AR encouraging creation; and (5) AR fostering transference. To investigate the effects of design-based AR learning on students' behaviors in the development of CT, this study took AI Traffic Lights as the lesson's subject and required students to create traffic lights with intelligent sensing and transformations using Arduino hardware, software and graphical programming tools. The experimental group employed the design-based-AR learning approach, provided a design log, AR learning resources, and scoring rubrics as a scaffold, while the control group employed a conventional learning strategy provided learning task sheets, multimedia resources such as videos, audio and pictures, and the same scoring rubrics. Students in the experimental group, supported by AR resources, completed the preliminary traffic light model and had a discussion, analyzing its strengths and weaknesses through evaluation, thinking about the necessity of the design to achieve the goal, and finding the reasons and key gaps.

Guided by a scoring scale, each student scored and gave feedback on others' work. After completing the grading, the groups analyzed the limitations of the existing solutions in depth through cooperation and communication and optimized the solutions and works. The control group was primarily instructed by the teacher. The teacher created scenarios and demonstrated traffic lights with the support of videos and pictures, explaining to students the key issues and knowledge in traffic light design. After observing the teacher's operation and demonstration, students relied on the design scheme and the fixed template provided by the teacher to create intelligent traffic lights. Finally, they conducted peer-to-peer assessments and scored their work.

The design-based AR learning strategy in this study presents abstract knowledge in the form of more intuitive 3D models or virtual scenes by creating animated AR resources to enhance students' perception and understanding of the knowledge (Figure 1). The 3D model library covers 3D models of various hardware and software elements related to AI traffic lights, such as traffic light bodies, sensors, and signals, with a high degree of realism and detail. It can be operated by rotating, zooming and panning to observe the model structure in all directions. The virtual scene builds a virtual city traffic intersection scene, including roads, vehicles, pedestrians, pedestrian crossings and traffic lights. Students can observe the operation status of traffic lights in different traffic flow and pedestrian crossing conditions.

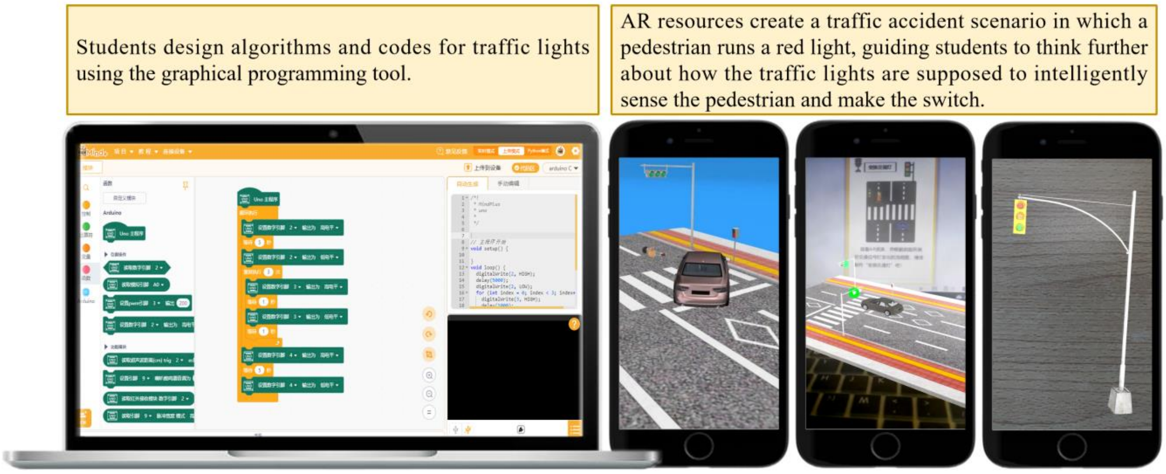


Figure 1. The functions and examples of design-based-AR learning approach.

3. Method

3.1. Participants

In the study, lag sequential analysis (LSA) was employed to analyze students' behaviors during the completion of the AI Traffic Light task. The study participants (*N*=98), were 46 female (46.9%) and 52 male (53.1%) sixth-grade students in southern China. All of these students were selected because they had previously participated in an AI teaching program for visualization tools and had some similar learning experiences. In this course, they were assigned to study groups of five students to ensure a consistent level of basic competence and initial knowledge in each group. We randomly divided 98 students into an experimental group and a control group, with 49 students in each of the two groups. All students were taught by the same teacher with extensive experience in teaching programming.

3.2. Data collection and analysis

To observe students' behaviors during the CT development of design-based-AR Learning, we recorded each student's behavior throughout the learning process using video. We first collected video data from the 1-hour normal course. Students were then asked to observe the AR resources and create two different AI traffic lights during the 1-hour session, documenting their projects according to the design log or task sheet provided. In addition, we use video to identify student behaviors that improve their work based on scoring rubrics and peer feedback.

Table 1. The coding table of learning behaviors for CT development.

Code	Content	Description
A	Abstracting the gap	Thinking about a problem from the whole perspective to identify gaps.
B	Examining the rubrics	Reading CT works scoring rubric.
C	Searching the Internet	Browsing the Web and reading the information on the Internet.
D	Decomposing	Breaking down CT problems into more manageable sub-problems to solve them.
E	Algorithming	Planning solutions to problems to write algorithmic codes.
F	Generalizing	Recognizing the solutions to specific problems and applying them to similar problems.
G	Evaluating	Finding the best solution given the state of available resources.
H	Re-algorithming	Optimizing the solution and rewriting codes with better algorithms for a problem.

The video data was captured via video recording software installed on the students' computers. We replayed the video files and encoded all the video data. LSA was conducted by using GSEQ 5.1 software to analyze the pattern of students' learning behaviors during the CT development activities. The CT behavior coding scheme was adapted from Tsai et al. (2021) and Lin et al. (2023), as shown in Table 1.

Video data was encoded by two researchers based on this coding scheme, and each researcher performed 15 seconds to 15 seconds of real-time data encoding. Both researchers were trained before coding and the differences in coding were discussed. The inter-rater kappa criterion of 0.864 was evaluated.

4. Results

This study examined the behavior patterns of students in the experimental and control groups in CT learning. It can be seen from Figure 2 that both groups performed similarly in C→C, A→D and G→F. It means that all students showed continuity in searching for information on the Internet (C→C). In addition, they exhibited two single sequences, i.e., from Abstracting the gap to Decomposing (A→D) and Evaluating to Generalizing (G→F).

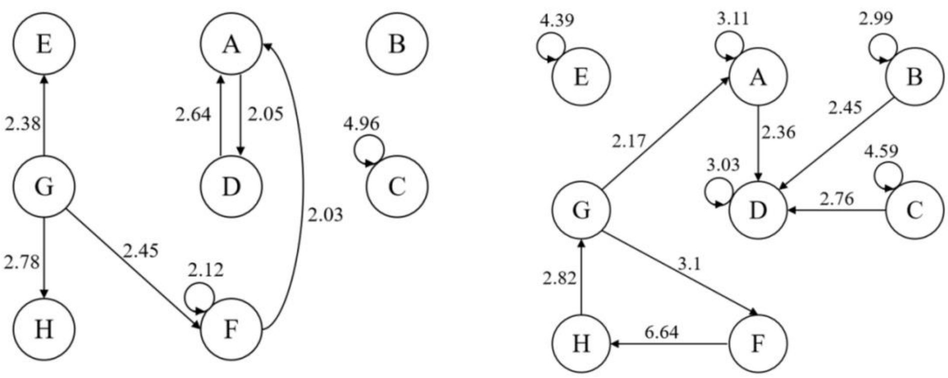


Figure 2. Behavior pattern diagram of control (left) and experimental (right) groups.

Students in the experimental group showed more algorithm behavior (E→E), as well as special sequences of G→F, F→H, and H→G. In the CT development activity, students in the experimental group behaved more frequently with generalization. They generalized after evaluating (G→F), reconstructed the algorithm via generalizing (F→H) and evaluated it again (H→G). The control group, on the other hand, showed sequences of re-algorithm (G→H), continuing algorithm (G→E) and generalizing (G→F) after the evaluation.

In addition, the experimental group exhibited a special sequence of B→B, B→D, C→D, and D→D. Their decomposition behaviors were more frequent. They decomposed after repeatedly examining the rubrics (B→B, B→D), as well as after searching for information on the Internet (C→D), and the decomposition behavior was recurrent (D→D). In contrast, the control group did not show similar behaviors.

5. Discussion

This study proposed a design-based-AR learning approach to investigate the effects of the approach on students' behavior during the development of CT. In terms of the repeating sequence E→E, the students in the experimental group repeated the algorithm design several times during the coding process. They continuously deepened the comprehension of computational concepts, logical structures and problem solving methods, and gradually developed a more explicit and structured framework of CT to optimize and adjust the code to improve the solution. In terms of the special sequence of G→F→H→G→A, students summarized their knowledge and existing experience after evaluating the original solution, redesigned the algorithm, evaluated the feasibility of the proposed solution again, and checked the limitations of the solution to better break down the task into easily handled sub-tasks. This pattern of behavior suggests that students, based on evaluation and feedback, examine the problems from different perspectives and are better prepared to transfer their existing learning experiences to new contexts finding more optimal algorithms and solutions when faced with similar problems. These findings are consistent with the results of Lin et al. (2023), which also showed the positive effect of mind visualization tools on CT development. In contrast, the control group behaved differently, after the evaluation, they preferred to reconstruct the algorithm directly (G→H), continue the algorithm (G→E) or generalize (G→F). This means that although they improved the algorithm quickly, they may have neglected the possibility of further exploring other solutions and considering similar alternative situations, limited to a single solution.

We also found that the experimental group of students exhibited the B→B sequence, i.e., the behavior of repeatedly examining the rubrics. They paid more attention to the repeated examination of the scoring criteria when producing the AI traffic light works to grasp the objectives of the task more accurately, and then adjust their thinking and behavior. In addition, they develop decomposition behaviors after abstracting the gap (A→D), examining the rubrics (B→D), reading the information on the Internet (C→D), or decomposing the problems (D→D). In design-based-AR learning, the CT development activities enable students to confront complex problems and concepts to capture the problem on the macro level and dynamically prioritize the solution by further breaking it down into operational sub-tasks. These findings are similar to the results of Yang et al. (2023), which also suggested that AR resources providing 3D visualizations contribute to students' problem decomposition skills.

This study reveals that the integration of CT development activities with visualization contexts provides students with visual thinking guidance, which better helps them understand the complicated concepts of computational thinking and improves their CT learning behaviors. It provides pedagogical guidelines for CT educational practices.

References

Kandemir, C. M., Kalelioğlu, F., & Gülbahar, Y. (2021). Pedagogy of teaching introductory text‐based programming in terms of computational thinking concepts and practices. *Computer Applications in Engineering Education, 29*(1), 29-45.

Lin, X.-F., Wang, J., Chen, Y., Zhou, Y., Luo, G., Wang, Z., Liang, Z.-M., Hu, X., & Li, W. (2023). Effect of a Reflection-Guided Visualized Mindtool Strategy for Improving Students' Learning Performance and Behaviors in Computational Thinking Development Development. *Educational Technology & Society, 26*(2), 165-180.

Tsai, M. J., Liang, J. C., & Hsu, C. Y. (2021). The computational thinking scale for computer literacy education. *Journal of Educational Computing Research, 59*(4), 579-602.

Wong, G.K.-W., & Cheung, H.-Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments, 28*(4), 438-450.

Yang, F. C. O., Lai, H. M., & Wang, Y. W. (2023). Effect of augmented reality-based virtual educational robotics on programming students' enjoyment of learning, computational thinking skills, and academic achievement. *Computers & Education, 195*, 104721.

Yang, S., Zhu, S., & Qin, W. (2025). Enhancing computational thinking assessment: A multimodal cognitive diagnostic approach. *Thinking Skills and Creativity, 56*, 101752.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Self-Demand vs. Time Availability: A Sociological Analysis of STEM Project Performance Among School and University Students in the Arequipa Region of Peru

Jair Jesús León LUCANO[1*], Juan Diego Cerrón SALCEDO[2], Daniel Haro LAQUE[3], Alberto Torres HISNOSTROZA[4]
[1,2,3,4]Continental University, fabrication laboratory area, Peru/Arequipa
jleonl@continental.edu.pe, jcerrons@continental.edu.pe, dharo@continental.edu.pe, atorresh@continental.edu.pe

Abstract: *This study examines how self-demand and time availability influence STEM project performance among school and university students in Arequipa, Peru. Using mixed methods, 14 projects were evaluated, revealing that school students outperformed university peers (average scores: 73.1 vs. 65.1), largely due to structured support and fewer external commitments. The findings highlight the sociological impact of institutional and personal constraints, recommending balanced workloads, enhanced mentorship, and better resource allocation to improve STEM learning outcomes across education levels. The study concludes that strategies emphasizing balanced workloads, targeted mentorship, and improved resource allocation are essential for optimizing STEM project-based learning across different educational stages.*

Keywords: STEM education, self-demand, time availability, sociological analysis, project-based learning

1. Introduction

STEM education plays a critical role in modern academic curricula, integrating science, technology, engineering, and mathematics to equip students with essential problem-solving and analytical skills (Kim et al., 2015). The increasing demand for professionals in STEM fields has led to the widespread adoption of educational strategies aimed at enhancing student engagement and competency development (Soto, 2020). Among these strategies, project-based learning (PBL) has proven particularly effective, fostering creativity, collaboration, and hands-on experience in real-world applications (Jara et al., 2011). Differences in institutional support, curriculum, mentorship, and socio-economic conditions create disparities in how students at various academic levels engage with and benefit from STEM education (Hsueh & Kuo, 2016).

Furthermore, the availability of time and the level of self-demand significantly impact student engagement and performance in STEM projects. Academic pressure, extracurricular obligations, and external commitments can either enhance or hinder a student’s ability to focus on project-based learning (Czerniewicz & Brown, 2014). Understanding these dynamics is crucial for optimizing educational frameworks that support students across different levels of learning (Levy & Schady, 2013).

The paper examines STEM education, project-based learning, and performance factors, offering insights on improving student outcomes through balanced workloads, strong mentorship, and curriculum adjustments to ensure both theoretical and practical skill development (Montés et al., 2023).

2. Methodology

A sequential exploratory mixed-methods design was employed, integrating a quantitative component (evaluation of projects using a standardized rubric) and a qualitative component (semi-structured interviews and/or open-ended questionnaires) (Hernández-Sampieri, 2010). The sociological approach (Sampieri et al., 1998) focuses on understanding how contextual factors—particularly self-management (or "self-demand") and time availability—affect students' performance in STEM projects.

2.1. Sample selecting

The study involved 14 purposively selected STEM project teams—7 from secondary schools (ages 13–16) and 7 from universities (ages 19–22) in Arequipa. Each team met criteria including a STEM focus, voluntary participation with institutional approval, and project completion within 1 to 4 months.

2.2. Project evaluation

The rubric was adapted from international PBL and STEM standards, incorporating criteria such as:

- Clarity and relevance of the problem to be solved.
- Methodological design and creativity in the solution.
- Implementation and validation of results.
- Quality of documentation, references, and final presentation.

Each criterion was scored on a scale from 0 to 10. The final score (maximum of 100) was obtained from the weighted sum of the individual items.

3. Results

For the quantitative analysis, descriptive measures (means, standard deviations) were calculated for the rubric scores.

Mean comparison tests (independent samples t-test) were conducted to identify statistically significant differences in performance between the secondary school and university teams.

The significance level was set at $p < .05$ (see Tables 1, and 2)

Table 1. School projects

	Average	Standard deviation	Scores
School	73.1	2.4	76, 70, 69.5, 73, 74.5, 73, 76

Table 2. University projects

	Average	Standard deviation	Scores
University	65.1	2.4	67, 66, 63.5, 65.5, 64, 65.5, 64.5

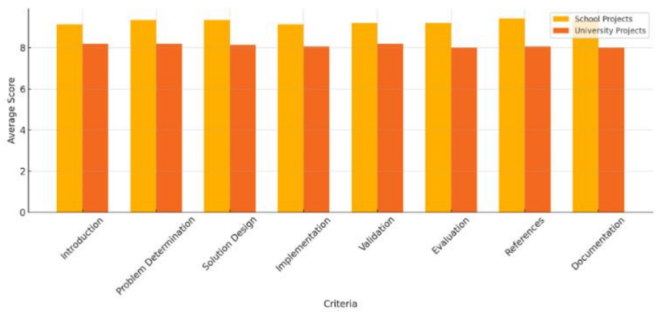


Figure 1. School vs. University Projects

Figure 1 compares the average scores of school and university projects for each evaluation criterion, school projects outperform university projects across all criteria, with consistently higher scores in categories such as introduction, problem determination, solution design, and documentation.

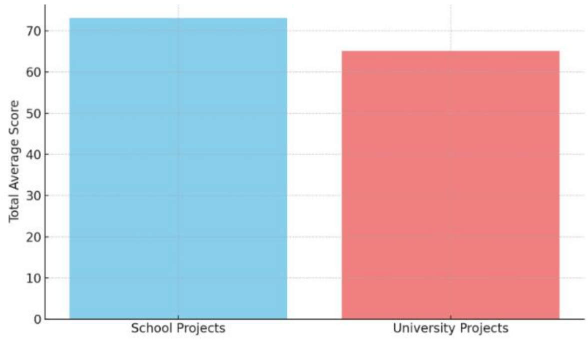


Figure 1. Total Average Scores

Figure 1 presents the overall performance of the two groups, showing that school projects had a significantly higher total average score (73.1) compared to university projects (65.1).

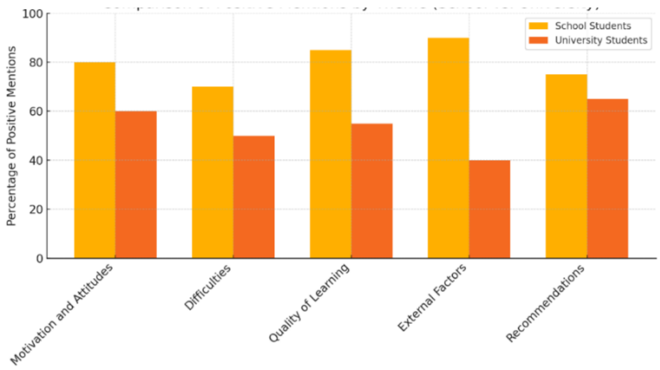


Figure 2. Comparison of positive mentions by theme (school vs. university)

Figure 2 compares the percentage of positive mentions across key themes (Motivation and Attitudes, Difficulties, Quality of Learning, External Factors, and Recommendations) between school and university students.

School students consistently report more positive mentions across all themes, particularly in:

- Motivation and Attitudes (80% vs. 60%): School students feel more engaged and supported.
- External Factors (90% vs. 40%): Schools provide more resources and parental support compared to university environments.

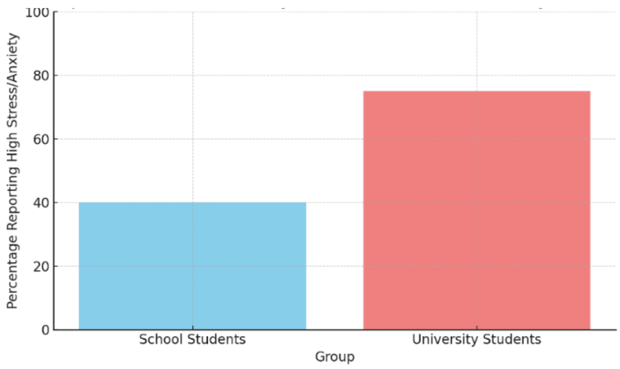


Figure 3. Comparison of positive mentions by theme (school vs. university)

Figure 3 compares University students experience significantly higher stress (75%) during STEM projects than school students (40%), due to academic overload, part-time jobs, and limited guidance. In contrast, school students benefit from structured support and fewer external pressures, underscoring the need for interventions to reduce stress in higher education.

4. Conclusions

This study highlights a paradox in STEM project performance: although university students possess greater technical knowledge, their output is hindered by stress, academic overload, and part-time work. In contrast, school students—despite their lower technical expertise—achieve higher scores due to structured support and focused learning environments.

Quantitative results show school teams averaged 73.1 on a standardized rubric, outperforming university teams, who averaged 65.1. This superiority extended across various criteria, including documentation and project quality. Qualitative data revealed that school students displayed higher motivation and engagement, while university students faced fragmented schedules and external demands.

From a sociological lens, the lower performance of university students is linked to systemic issues in higher education, where survival often trumps creativity. Conversely, school students benefit from consistent guidance and fewer external pressures. This disparity is reinforced by stress data: 75% of university students reported high stress, versus 40% of school students, largely due to lack of mentorship and time constraints.

The findings call for targeted policy changes in higher education to overcome structural barriers impacting STEM project performance. Key recommendations include reducing academic overload through interdisciplinary projects, strengthening mentorship to mirror school-level support, offering financial aid for students with part-time jobs, and promoting stress management via flexible deadlines and mental health services. These measures aim to foster an environment where students can effectively apply their technical skills to real-world challenges, enhancing overall STEM learning outcomes.

References

Czerniewicz, L., & Brown, C. (2014). The habitus and technological practices of rural students: a case study. *South African Journal of Education*, *34*(1).

Hernández-Sampieri, R. (2010). *Fundamentos de Metodología de la Investigación*. McGraw-Hill Interamericana de España S.L. https://books.google.com.pe/books?id=WILJNAAACAAJ

Hsueh, S.-L., & Kuo, C.-H. (2016). Factors in an interdisciplinary curriculum for the students of industrial design designing multifunctional products. *Eurasia Journal of Mathematics, Science and Technology Education*, *12*(4), 1075–1089.

Jara, C. A., Candelas, F. A., Puente, S. T., & Torres, F. (2011). Hands-on experiences of undergraduate students in Automatics and Robotics using a virtual and remote laboratory. *Computers & Education*, *57*(4), 2451–2461.

Kim, C., Kim, D., Yuan, J., Hill, R. B., Doshi, P., & Thai, C. N. (2015). Robotics to promote elementary education pre-service teachers' STEM engagement, learning, and teaching. *Computers & Education*, *91*, 14–31.

Levy, S., & Schady, N. (2013). Latin America's social policy challenge: Education, social insurance, redistribution. *Journal of Economic Perspectives*, *27*(2), 193–218.

Montés, N., Zapatera, A., Ruiz, F., Zuccato, L., Rainero, S., Zanetti, A., Gallon, K., Pacheco, G., Mancuso, A., & Kofteros, A. (2023). A novel methodology to develop STEAM projects according to national curricula. *Education Sciences*, *13*(2), 169.

Sampieri, R. H., Collado, C. F., & Lucio, P. B. (1998). *Metodología de la investigación*. McGraw-Hill. https://books.google.com.pe/books?id=e4OnuQAACAAJ

Soto, D. A. (2020). *Technology and the future of work in emerging economies: What is different*.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

The Application and Effectiveness of Educational Robotics in Elementary Programming and Technological Humanistic Literacy Development

Chia-Yen FENG*, Ming-Puu CHEN, Li-Chun WANG
National Taiwan Normal University, Taiwan
yeni0412@gmail.com, mpchen@ntnu.edu.tw, cct101wang@gmail.com

Abstract: *This study explores how educational robotics enhances elementary students' computational thinking, problem-solving skills, and technological humanistic literacy. A total of 23 sixth-grade students participated in a seven-week Kebbi Air robot programming course. A computational thinking competency test and a problem-solving self-assessment questionnaire were used for quantitative analysis, and student interviews were conducted to examine their perspectives on technological humanistic literacy. The results indicated significant improvements in students' understanding and application of computational thinking concepts, particularly in conditional logic and iteration structures. In problem-solving, the "testing, evaluation, and improvement" dimension showed the most notable progress. Interview findings revealed that students were not only interested in robotics technology but also considered its societal implications. This study validates the potential of educational robotics as a medium for programming education and a catalyst for fostering students' awareness of technology's societal impact.*

Keywords: Elementary Programming Education; Educational Robotics; Computational Thinking; Technological Humanistic Literacy

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

機器人教育在小學程式設計與科技關懷素養的應用與成效探討

豐佳燕*，陳明溥，王麗君
國立臺灣師範大學，臺灣
yeni0412@gmail.com，mpchen@ntnu.edu.tw，cct101wang@gmail.com

【摘要】本研究探討機器人教育如何有效提升國小學生的運算思維、問題解決能力及科技關懷素養。23 名六年級學生參與為期七週的機器人程式設計課程，透過運算思維能力測驗與問題解決能力自評表進行量化分析，並訪談學生科技關懷體驗。研究結果顯示，學生在運算思維概念理解與應用能力皆顯著提升，在問題解決「測試、評鑑與改進」表現尤為突出。訪談結果亦指出，學生們不僅對機器人技術本身感興趣，也思考機器人對社會的影響。本研究驗證機器人教育不僅能作為程式設計學習媒介，更能啟發科技關懷素養的潛力。

【關鍵字】國小程式教育；機器人教育；運算思維；科技關懷

1. 前言

運算思維（Computational Thinking, CT）被視為 21 世紀關鍵能力，包含分解、模式辨識、抽象化與演算法思維（Wing, 2006; Brennan & Resnick, 2012）。機器人教育（Educational Robotics, ER）已廣泛應用於中小學教育，能有效提升學生的邏輯推理與問題解決能力（Zhang et al., 2021; Hong, 2024）。此外，機器人應用於長者照護、視障輔助與智慧醫療等場域，顯示其在人文關懷上的潛力（Rojas et al., 2025）。

凱比（Kebbi Air）機器人具備語音與視覺辨識、互動溝通與可程式化特性，適用於培養學生運算思維與科技素養。本研究探討機器人教學對國小學生學習成效的影響，並評估其在運算思維、問題解決與科技關懷素養方面的成果。

2. 文獻探討

2.1. 運算思維與問題解決

近年來的研究顯示，透過圖形化程式設計環境與機器人實作，學生能夠直觀地學習程式邏輯，並透過即時回饋調整錯誤，從而提升運算思維與問題解決能力（Grover & Pea, 2018; Zhang et al., 2021）。此外，機器人學習活動能夠強化學生對運算思維核心概念的掌握，尤其在條件判斷與迴圈結構方面表現尤為顯著（Hong, 2024）。

Chang 等人（2020）發現，透過積木式程式設計結合機器人教學，學生得以即時進行驗證與除錯，進而促進抽象邏輯與問題拆解能力，並增強後設認知發展。Souza et al.（2021）融入小組合作與專題探究模式，則能更全面地提升學生的問題解決能力（Negrini et al., 2023）。

2.2. 機器人教育的多元應用

機器人教育已廣泛應用於中小學課程，並在運算思維、問題解決能力與學習動機等方面展現顯著成效（Zhang et al., 2021；Hong, 2024）。許多研究指出，機器人學習環境能提供即時回饋，使學生在操作過程中強化程式邏輯、錯誤調適與演算法應用能力（Grover & Pea, 2018; Negrini et al., 2023）。透過專題導向學習（Project-Based Learning, PBL），學生不僅能深化運算思維，亦能培養自主學習與協作能力（Valls Pou et al., 2022）。

機器人教育的影響不僅限於 STEM 領域，其對於學生的創造力、批判性思維與人文關懷意識亦具有正向影響（Atman Uslu et al., 2023）。透過設計以解決真實問題為導向的機器人專題，如長者照護、視障輔助與智慧醫療，學生能夠學習如何將科技應用於社會關懷，進一步

培養科技倫理與社會責任感（Li, 2020; Rojas et al., 2025）。此外，機器人學習環境的互動特性，使學生更容易理解科技在人類社會中的角色，並提升數位幸福感（Negrini et al., 2023;Ching & Hsu, 2024）。本研究透過凱比機器人課程，期望學生在學習程式設計的同時，亦能提升對科技關懷的理解，進一步探索科技與社會發展的關聯性。

3. 研究方法

3.1. 研究對象

本研究對象為臺灣某國小六年級資優班學生，共 23 名（男 12 人，女 11 人），年齡介於 11 至 12 歲。參與學生皆具備基礎 Scratch 程式設計經驗，但無機器人操作與程式控制經驗。本研究於課程開始前進行程式先備知識測驗，以確認學生起始能力無顯著差異，確保研究結果的信效度。

3.2. 研究設計

本研究採準實驗研究法，透過前測-後測設計評估機器人課程對學生的學習成效。研究期間為七週，共 280 分鐘，學生透過凱比機器人學習程式設計、機器人控制與專題應用，並透過專題任務發展問題解決能力與科技關懷素養。

3.3. 教學內容

本研究依據專題導向學習設計三大教學單元，課程內容如表 1 所示。

表 1 機器人專題課程設計與學習目標

單元	課程內容	學習目標	課程時間
單元一： 凱比初探	介紹機器人基本功能、語音與視覺辨識、程式介面操作（圖 1）。	熟悉操作機器人，學習語音互動與感測器運用。	60 分鐘
單元二： 運算思維與問題解決	設計控制機器人，學習條件判斷、迴圈與變數應用（圖 2）。	培養運算思維，能利用程式解決特定問題。	100 分鐘
單元三： 科技關懷專題	設計機器人應用於長者陪伴、幫助弱勢同學等社會關懷專題。	提升科技關懷意識，強化問題解決能力	120 分鐘

圖 1 學生與機器人互動

圖 2 學習以積木程式控制機器人

3.3. 研究工具

3.3.1. 運算思維能力測驗

運算思維能力測驗旨在評估學生在課程前後對運算思維概念的理解與應用能力。本測驗參考 Brennan 與 Resnick（2012）提出的運算思維架構進行修訂，依認知層次分為兩大向度：概念理解為評估學生對積木程式指令的認知程度；應用能力則測量學生是否能運用正確的指令組合解決問題。整體測驗的 Cronbach's α 值為 0.73，顯示具有良好的內部一致性信度。

3.3.2.凱比專題問題解決能力自評表

本研究採用問題解決能力自評表評估學生於凱比機器人專題製作歷程的問題解決能力。本問卷參考柯淇羚（2020）建構的科技問題解決能力，並針對本研究的教學內容進行適當修

正，訂定 21 題，涵蓋問題識別、資訊蒐集、解決方案設計、測試與評鑑、優化與反思等五大向度。問卷採用五點李克特量表（Likert Scale, 1-5）。本量表之 Cronbach's α 值為 0.81，顯示具有高度信度。

4. 研究結果與討論

4.1. 運算思維能力

為探討機器人課程對學生運算思維能力的影響，本研究對學生進行前後測測驗，結果如表 2 所示。學生在「概念理解」與「應用能力」兩個向度均有顯著提升，其中「應用能力」的平均差異最高（$M = 2.41$, SD = 0.85, $t_{(22)} = 3.92$, $p = .04$），顯示學生在實作與運算思維遷移能力上獲得較大成長。此外，訪談回饋顯示，學生認為透過機器人互動與即時測試，有助於強化對程式結構的掌握，提升邏輯思維與問題解決能力。

表 2 運算思維能力測驗前後測結果

向度	前測 *M*(SD)	後測 *M*(SD)	平均差異	*t*	*p*
概念理解	5.82(1.21)	7.65(1.03)	1.83 (0.79)	3.74	.02
應用能力	6.13(1.15)	8.54(1.92)	2.41(0.85)	3.92	.04
總分	11.95(2.03)	16.19(1.95)	4.24(1.36)	4.21	.03

4.2. 機器人專題問題解決能力

為進一步評估學生在凱比機器人專題製作歷程中的問題解決能力發展，本研究透過問題解決能力自評表與教師評估進行分析，結果如表 3 所示。學生自評與教師評估的結果大致相符，尤以「測試與評鑑」與「優化與反思」向度提升最為明顯，顯示學生在凱比機器人專題製作過程中，能夠透過實驗測試來評估方案的可行性，並根據反思進行修正。訪談回饋亦指出，部分學生在專題過程中遇到程式錯誤時，能主動分析錯誤來源，並透過與同學討論找到最佳解決方案，展現高度的問題解決能力與協作學習精神。

表 3 凱比專題問題解決能力自評與教師評估結果

向度	學生自評 *M* (SD)	教師評估 *M* (SD)	平均差異	*t*	*p*
問題識別	3.82 (0.71)	3.95 (0.65)	0.13 (0.42)	1.12	0.27
資訊蒐集	3.65 (0.78)	3.79 (0.70)	0.14 (0.48)	1.04	0.29
解決方案設計	3.74 (0.69)	3.88 (0.73)	0.14 (0.51)	1.08	0.28
測試與評鑑	4.12 (0.82)	4.01 (0.77)	-0.11 (0.39)	1.35	0.19
優化與反思	4.25 (0.76)	4.18 (0.71)	-0.07 (0.35)	1.21	0.24
總分	3.92 (0.74)	3.96 (0.71)	0.04 (0.38)	1.16	0.26

4.3. 科技關懷素養

本研究透過訪談探討學生在執行專題任務時，對機器人在社會應用與人文關懷的看法。

(1) 機器人應用於社會服務：多數學生認為，機器人能幫助視障者導引方向、提醒長者用藥，或輔助學習困難的同學。

(2) 科技倫理與責任：部分學生關注機器人可能帶來的倫理問題，如取代人類工作或影響人際關係。

(3) 人機互動的未來發展：部分學生期待機器人具備情感回饋功能，例如陪伴獨居長者或提供心理支持。

多數學生在程式實作中逐步發展出對於科技應用於社會的關懷意識，這與呂昌育（2024）關於數位幸福感與兒童心理健康的研究相呼應。

5. 結論與未來展望

本研究探討機器人教育如何提升國小學生的運算思維、問題解決能力及科技關懷素養。結果顯示，透過機器人課程，學生在運算思維的概念理解與應用能力皆有顯著提升。此外，問題解決能力於「測試與評鑑」及「優化與反思」向度的成長尤為顯著，顯示專題導向學習與動手實作有助於提升學生的問題解決能力。訪談結果顯示，學生開始關注機器人在社會服務領域的應用並對科技倫理與社會責任產生初步思考。未來可進一步擴展機器人課程的跨學科應用，結合語文、社會或藝術領域，以強化學生對科技應用的理解與創造力。

參考文獻

呂昌育。 (2024)。**探索孩童幸福感之家庭因素：結合數位表現型與生態瞬時評估法**〔未出版之碩士論文〕。國立臺灣師範大學。

柯淇羚（2020）。**國中生科技問題解決歷程與能力指標之建構研究**〔未出版之碩士論文〕。國立臺灣師範大學。

Atman Uslu, N., Yavuz, G. Ö., & Koçak Usluel, Y. (2023). A systematic review study on educational robotics and robots. *Interactive Learning Environments*, *31*(9), 5874-5898. https://doi.org/10.1080/10494820.2021.2023890

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, Canada (Vol. 1, p. 25). http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

Ching, Y. H., & Hsu, Y. C. (2024). Educational robotics for developing computational thinking in young learners: A systematic review. *TechTrends*, *68*(3), 423-434. https://doi.org/10.1007/s11528-023-00841-1

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. *Computer Science Education: Perspectives on Teaching and Learning in School*, *19*(1), 19-38.

Hong, L. (2024). The impact of educational robots on students' computational thinking: A meta-analysis of K-12. Education and Information Technologies, 29(11), 13813-13838. https://doi.org/10.1007/s10639-023-12415-y

Negrini, L., Giang, C., Bonaiuti, G., Cascalho, J. M., Primo, T. T., & Eteokleous, N. (2023, March). Educational robotics as a tool to foster 21st century skills. In *Frontiers in Education* (Vol. 8, p. 1186029). Frontiers Media SA. https://doi.org/10.3389/feduc.2023.1186029

Rojas, E. M., Valencia-Arias, A., Coronado, M. H. V., Gamarra, J. M. B., Agudelo-Ceballos, E., Benjumea-Arias, M. L., & Mori, L. V. (2025). Educational robotics for primary education: An analysis of research trends. *Eurasia Journal of Mathematics, Science and Technology Education, 21*(3), em2602. https://doi.org/10.29333/ejmste/16050

Valls Pou, A., Canaleta, X., & Fonseca, D. (2022). Computational thinking and educational robotics integrated into project-based learning. *Sensors, 22*(10), 3746. https://doi.org/10.3390/s22103746

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Zhang, Y., Luo, R., Zhu, Y., & Yin, Y. (2021). Educational robots improve K-12 students' computational thinking and STEM attitudes: Systematic review. *Journal of Educational Computing Research, 59*(7), 1450-1481. https://doi.org/10.1177/0735633121994070

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

The Computational Thinking Performance of Taiwanese Elementary School Students on Bebras Challenge Tasks on ViLLE

Tai-Ping HSU[1], Shih-Hua HUANG[2], Ting-Chia HSU[3*], Valentina DAGIENĖ[4]
[1,2,3] National Taiwan Normal University, Taiwan
[4] Vilnius University, Lithuania
ckhsu@ntnu.edu.tw

Abstract: *In the digital age, computational thinking (CT) is increasingly essential, yet existing assessment tools, especially for interdisciplinary CT skills, remain inadequate, limiting integration into school curricula. This study addressed the issue by designing an instructional activity for fifth-grade students, combining the ViLLE online platform with Bebras tasks, which embed CT concepts through real-world scenarios and offer personalized practice with real-time feedback. Results showed strong student performance, with an average accuracy of 94.98%, though slightly lower on tree-structured problems. The integration effectively enhanced computational problem-solving, particularly in "Little House," "Teeth," and "Beaver" tasks. This study provides preliminary evidence of the feasibility and effectiveness of using ViLLE and Bebras tasks in elementary CT education, highlighting their potential for assessing and improving CT skills. Future research should expand the sample and explore the influence of different question types on CT development.*

Keywords: Computational Thinking, Bebras Challenge, Online Learning Platform

1. Introduction

In today's digital society, computational thinking (CT) has become a vital 21st-century skill (Lai & Wong, 2022). With computational methods widely applied across disciplines, integrating CT into school curricula is essential (Lodi & Martini, 2021). Studies have highlighted programming as an effective tool to develop CT by enhancing students' cognitive abilities (Lai & Wong, 2022). Even unplugged programming activities can improve CT skills (Dağ et al., 2023).

However, CT assessment tools remain insufficient (Guggemos et al., 2023), lacking effective measures to accurately evaluate students' CT abilities, which hinders curriculum integration (Chan et al., 2021). To address this, digital tools, games, and programming languages combined with formative or summative assessments offer more comprehensive evaluations (Chan et al., 2023). Yet, challenges persist, as many assessments focus narrowly on computer science, overlooking interdisciplinary needs (Lu et al., 2022; Chan et al., 2021).

Bebras Computational Thinking Challenge tasks provide a promising interdisciplinary assessment method by embedding real-life contexts and diverse question formats, fostering both CT skills and learning motivation (Lonati, 2020). Online platforms further enhance Bebras tasks through flexibility, personalized practice, real-time data collection, and immediate feedback (Hooshyar et al., 2021).

This study addresses CT assessment challenges by integrating Bebras tasks into the ViLLE online platform, offering personalized practice and real-time feedback. As preliminary research, it explores the feasibility of this approach for elementary CT education. Focusing on fifth-grade students, the study analyzes their Bebras task performance within ViLLE, using platform data to understand student interactions and problem-solving behavior.

2. Literature Review

2.1. Methods for Assessing Computational Thinking

The primary goal of CT assessment is to evaluate individuals' ability to apply computational concepts and problem-solving strategies, with methods rapidly evolving (Lu et al., 2022). Research has developed various approaches, such as CT tests for proficiency (Chan et al., 2021) and combining performance-based tests with self-assessment scales for comprehensive evaluation (Guggemos et al., 2023), where self-assessments offer metacognitive insights (Jiang & Li, 2021). While multi-method assessments enhance understanding of CT abilities, challenges remain—most tools still focus on computer science with limited interdisciplinary applicability (Lu et al., 2022; Chan et al., 2021). To address this, Bebras Computational Thinking Challenge tasks offer a solution by embedding real-world scenarios and diverse question formats to assess problem-solving skills while boosting learning motivation (Lonati, 2020).

2.2. Bebras Computational Thinking Challenge Tasks

Bebras tasks are unplugged activities designed to develop CT skills by engaging students with informatics concepts through small problem-solving tasks that embed fundamental computer science principles, without requiring computers (Lonati, 2020; Kalelioğlu et al., 2022). Widely adopted in education, they have been used in countries like Turkey to assess CT across age groups and integrated into teacher training to promote CT instruction (Kalelioğlu et al., 2022; Lonati, 2020). Studies confirm their effectiveness in enhancing specific CT skills and evaluating problem-solving proficiency through student responses (Zapata-Cáceres et al., 2024; Kwon et al., 2021). However, limitations remain, such as the need to improve assessments across age groups, task types, and CT dimensions, as well as challenges teachers face in understanding and applying these tasks in daily teaching (Kalelioğlu et al., 2022; Lonati, 2020). Online platforms can address these issues by providing flexible, adaptive practice, real-time data collection, and personalized feedback to support student progress and targeted improvement.

3. Research Method

3.1. Research Design

This study adopted a data analysis approach to examine students' performance on Bebras Computational Thinking Challenge tasks (Bebras tasks) after participating in computational thinking (CT) instructional activities incorporating the ViLLE online learning platform. The study was conducted in a formal classroom setting, where the primary objective of the instructional activities was to enhance students' CT skills. The participants included five fifth-grade elementary school students (aged 11). The instructional activities spanned two weeks, with one session per week, each lasting 80 minutes.

For data analysis, students' performance on Bebras tasks within the ViLLE platform was examined post-instruction to assess their CT proficiency. The collected research data consisted of students' scores on Bebras tasks, which were obtained via the teacher's account backend on the ViLLE platform.

3.2. Research Instruments

This study utilized the ViLLE educational platform, a web-based multi-functional learning platform developed by the University of Turku, Finland. ViLLE offers various features, including practice exercises, learning analytics, and gamification modules.For this study's instructional activities, two primary modules of the ViLLE platform were employed:Practice Module and Learning Analytics Module.This study used Bebras tasks from the 2024–2025 cycle to evaluate students' CT abilities, integrating real-world scenarios to boost learning motivation and encourage CT application. A total of 41 tasks covering six themes were selected, with difficulty levels tailored to fifth-grade cognitive development for effective CT assessment. The tasks were designed to incorporate core CT components such as sequencing, selection, iteration, pattern recognition, algorithmic design, and logical reasoning.

3.3. Research Procedure

The instructional activities were conducted twice during regular class sessions, with each session lasting 80 minutes. The primary objective of the instructional activities was to enhance the computational thinking (CT) skills of fifth-grade students. The overall instructional process was structured as follows:

- CT Concept Explanation (20 minutes): At the beginning of each session, the teacher introduced fundamental CT concepts and their importance to the students.
- Bebras Task Practice (60 minutes): After the conceptual explanation, students engaged in hands-on practice by solving Bebras tasks on the ViLLE platform, guided by the teacher.

4. Research Results

4.1. Analysis of Bebras Task Performance – Overall Performance

To evaluate students' overall performance on Bebras tasks, their responses across all tasks were analyzed. Table 1 presents the descriptive statistics of students' overall performance. The results indicate that students achieved an average accuracy rate of 94.98%, with an average response time of 541.70 seconds. These findings suggest that students performed exceptionally well during Bebras task practice.

Table 1. Descriptive Statistics of Students' Overall Performance on Bebras Tasks.

Indicator	Mean	Standard Deviation
Average Accuracy Rate	94.98%	8.71%
Average Response Time	541.7 sec	362.77 sec

4.2. Analysis of Bebras Task Performance – Performance Across Different Task Types

To further examine the impact of different Bebras task types on students' performance, this section compares students' average accuracy rates and average response times across different task themes (Tree, Dam, Little House, Teeth, Tail, and Beaver). Table 2 presents students' performance across these task types.

Table 2. Students' Performance Across Different Bebras Task Types.

Task Type	Average Accuracy Rate	Average Response Time
Tree	77.25%	489.40 sec
Dam	96.51%	318.80 sec
Little House	100.00%	309.20 sec
Teeth	100.00%	568.80 sec
Tail	96.14%	802.20 sec
Beaver	100.00%	761.80 sec

Students demonstrated high accuracy across most task types, with Little House, Teeth, and Beaver tasks reaching 100%, showing strong mastery. Dam and Tail tasks also performed well, with accuracy rates of 96.51% and 96.14%. In contrast, the Tree task had a lower accuracy rate of 77.25%, suggesting it was more challenging for students.

5. Discussion and Conclusion

This study investigated fifth-grade students' performance on Bebras Computational Thinking Challenge tasks (Bebras tasks) after integrating the ViLLE online learning platform into CT instruction. With an overall average accuracy rate of 94.98%, the results suggest that practicing Bebras tasks through ViLLE effectively enhanced students' problem-solving abilities.Several factors may explain this strong performance. Bebras tasks embed CT concepts into real-life scenarios, making them more engaging than traditional abstract exercises (Lonati, 2020). Additionally, ViLLE's real-time feedback allowed students to promptly identify and correct misconceptions, fostering continuous improvement. Prior research supports the effectiveness of unplugged activities in developing CT skills (Dağ et al., 2023) and highlights the benefits of immediate feedback in online learning environments (Hooshyar et al., 2021). The combination of these features likely created an optimal environment for CT development.

However, consistent with previous findings on task difficulty variation (Kalelioğlu et al., 2022), students showed lower accuracy on Tree tasks (77.25%), indicating challenges with specific CT concepts. In contrast, perfect scores in Little House, Teeth, and Beaver tasks suggest these were well-aligned with students' cognitive levels or effectively reinforced through instruction. This highlights the potential of ViLLE and Bebras tasks in strengthening areas such as iteration and algorithm design, aligning with Zapata-Cáceres et al. (2024), who emphasized Bebras' role in fostering CT competencies.

References

Chan, S.-W., Looi, C.-K., Ho, W. K., & Kim, M. S. (2023). Tools and approaches for integrating computational thinking and mathematics: A scoping review of current empirical studies. *Journal of Educational Computing Research*, *60*(8), 2036–2080. https://doi.org/10.1177/07356331221098793

Chan, S.-W., Looi, C.-K., & Sumintono, B. (2021). Assessing computational thinking abilities among Singapore secondary students: A Rasch model measurement analysis. *Journal of Computers in Education*, *8*(2), 213–236.

Dağ, F., Şumuer, E., & Durdu, L. (2023). The effect of an unplugged coding course on primary school students' improvement in their computational thinking skills. *Journal of Computer Assisted Learning*, *39*(6), 1902–1918.

Guggemos, J., Seufert, S., & Román-González, M. (2023). Computational thinking assessment – Towards more vivid interpretations. *Technology, Knowledge and Learning*, *28*(2), 539–568.

Hooshyar, D., Pedaste, M., Yang, Y., Malva, L., Hwang, G.-J., Wang, M., Lim, H., & Delev, D. (2021). From gaming to computational thinking: An adaptive educational computer game-based learning approach. *Journal of Educational Computing Research*, *59*(3), 383–409. https://doi.org/10.1177/0735633120965919

Kalelİoğlu, F., Doğan, D., & Gülbahar, Y. (2022). Snapshot of computational thinking in Turkey: A critique of 2019 Bebras challenge. *Informatics in Education*, *21*(3), 501–522. https://doi.org/10.15388/infedu.2022.19

Kwon, K., Cheon, J., & Moon, H. (2021). Levels of problem-solving competency identified through Bebras Computing Challenge. *Education and Information Technologies*, *26*(5), 5477–5498.

Lai, X., & Wong, G. K.-W. (2022). Collaborative versus individual problem solving in computational thinking through programming: A meta-analysis. *British Journal of Educational Technology*, *53*(1), 150–170.

Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. *Science & Education*, *30*(4), 883–908. https://doi.org/10.1007/s11191-021-00202-5

LONATI, V. (2020). Getting inspired by Bebras tasks. How Italian teachers elaborate on computing topics. *Informatics in Education*, *19*(4), 669–699. https://doi.org/10.15388/INFEDU.2020.29

Lu, C., Macdonald, R., Odell, B., Kokhan, V., Demmans Epp, C., & Cutumisu, M. (2022). A scoping review of computational thinking assessments in higher education. *Journal of Computing in Higher Education*, *34*(2), 416–461. https://doi.org/10.1007/s12528-021-09305-y

Zapata-Cáceres, M., Marcelino, P., El-Hamamsy, L., & Martín-Barroso, E. (2024). A Bebras computational thinking (ABC-Thinking) program for primary school: Evaluation using the competent computational thinking test. *Education and Information Technologies*, *29*(12), 14969–14998. https://doi.org/10.1007/s10639-023-12441-w

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Use of Computational Thinking Skills in Second Language Acquisition Among Adult Immigrants

Cintia TETELBOM SCHUCHMANN, Arnon HERSHKOVITZ*
Tel Aviv University, Israel
cintiat@tauex.tau.ac.il, arnonhe@tauex.tau.ac.il

Abstract: *This research explores the use of Computational Thinking skills in the process of learning a foreign language. We studied it in the context of new immigrants in Israel learning and using Hebrew. A qualitative approach was developed, and semi-structured interviews and think-aloud tasks were administered with 14 adult immigrants from diverse backgrounds. Findings demonstrate the prominence of Abstraction, Algorithms, and Debugging in the process.*

Keywords: Computational Thinking, Second Language Acquisition, Immigrant Education

1. Introduction

Second Language Acquisition (SLA) research examines the cognitive and social process of learning an additional language—aka second language (L2)—other than the mother tongue (Ellis, 1999; Kanwal et al., 2022). SLA is relevant in the case of immigrants, specially as there has been an increased migration pattern worldwide with an updated total estimated of over 280 million people that were living in a country different from the one where they were born (International Organization for Migration, 2024). Of these, 169 million of them (over 60%) were labor migrants; although labor migrants show high labor participation rates, it is highlighted that the rates have declined in the last decade, due to several factors, including insufficient language proficiency.

Learning a second language is the acquisition of a new mental representation used by others to communicate (Keating, 2016). That is, a new language is basically a new system to be added to the known structures. Therefore, it is important to investigate the cognitive mechanisms that support SLA and which often include domain-general skills that are also used in other areas of learning are employed.

Recently, there has been increased efforts to combine computational thinking (CT) beyond STEM disciplines. Researchers have explored its potential to support various learning processes, as CT skills can relate to problem-solving in every discipline, including language-related contexts (Bounou et al., 2023; Hsu et al., 2022; Yu et al., 2024). Research suggests that CT skills such as Decomposition, Pattern Recognition, and Debugging are actively used by second language learners (Liang, 2022; Sabitzer et al., 2018), which highlights the potential of integrating CT skills into language teaching to enhance language acquisition processes. Still, a deeper exploration of the CT skills used during SLA is needed, which is the gap that we bridge in this paper. The purpose of this research is to investigating how strategies that are used in learning a second language, specifically in the case of adult new immigrants, are related to CT skills.

2. Methodology

Population. The study focused on adult immigrants in Israel who have learned or are learning Hebrew as a second language in Israel. A total of 14 participants, aged 24 to 65, were selected based on their diverse linguistic, cultural backgrounds, different fields of occupation and methods of study. They were immigrants from Argentina, Brazil, Canada, Germany, Mexico, Russia, and USA; all had immigrated within the last five years and were at different proficiency levels in Hebrew.

Tools. A qualitative approach was employed, utilizing semi-structured interviews and think-aloud tasks. Immigrants L1 of which was English, Portuguese, or Spanish (12 of 14) were interviewed in these languages; otherwise, they were interviewed in English. The interview protocol was designed to extract insights into participants' thought processes while interacting with the Hebrew language and allow them to reflect on their learning process after immigration. Task-based activities were included to assess the application of the CT facets. Participants were asked to verbalize their thought process while performing tasks such as sentence structuring, error correction, and translation exercises.

Analysis. The unit of analysis was an interviewee's statement. Interview transcripts were coded with CT facets and skills (Shute et al., 2017), following the directed method (Hsieh & Shannon, 2005). A single statement could have been coded to multiple CT skills. Same-code statements were then aggregated to higher-level themes, in an iterative process that included both authors.

3. Findings

Overall, 571 statements related to CT were extracted from the interviews, ranging between 27-59 per participant (M=40.8, SD=10.7). We report here on the most prominent CT facets that can be linked to strategies used by participants.

3.1. Abstraction

Abstraction was by far the most prominent CT facet coded, related to about three-quarters of the statements.

Data Collection and Analysis. Participants analyzed the context to infer the meaning of new words, by finding other known words in the sentence. For example, Participant 1 mentioned scanning signs in the street to extract information without using an online translator. Another participant used the same skill when checking their emails in Hebrew: "[I] just scan it to understand the main words, to get 'okay, this is the meaning'" (P3). Participants used various contextual elements, e.g., "the other parts of the sentence and people's facial responses and emotions" (P6).

Pattern Recognition. In Hebrew, recognizing the morphology of a word, being its root or its structure, can be critical to understanding its meaning. Reading in Hebrew allows learners to access data linked to the letters used in the words and visualize the format, and listening provides vowel sounds, which can also help identify patterns: "I will at least try to see the roots of the word." (P12), "There are ways that you can try to understand. If certain letters are together, if you can pull the root from it or things of that nature" (P6).

Modeling. Modeling appears to be a fundamental strategy used by learners to express themselves in Hebrew. Participants reported using known verbs as models to conjugate unfamiliar verbs, relying on similarities in structure or sound. Participant 1 exemplifies: "I go by my models. That for each 'binyan' [linguistic structure] there is a model verb. And then, according to the time tense I want to conjugate, I follow the process.". Participant 4 explains that her modeling process is guided by the sonority when using a new verb: "I just say what sounds better in my head, I've been here four years. I think I kind of have a little bit of the ear you need to identify those concept kinds of things". Some participants mentally visualize conjugation verb tables to help them conjugate verbs correctly: "I think about what's the time that I need to say it, and then I remember, I actually can visualize how, for example, in the future, it's the letter Alef in the beginning" (P2).

3.2. Debugging

The Debugging facet was related to about a quarter of the statements. Participants reported frequently identifying their own mistakes and experiencing self-doubt when encountering errors. For example, Participant 2 said: "Sometimes when I read texts that I wrote in Hebrew for the second time then I can find mistakes that I made". Participant 1 corrected her pronunciation when reading during the interview, after continuing the sentence and only then understanding the correct pronunciation. Participants also were able to identify errors in others' use of Hebrew, especially other L2 speakers.

Participant 3 mentioned identifying “usual” mistakes, such as not conjugating a verb and saying it in the infinitive form in a sentence. Participant 4, when asked about recognizing mistakes, replied “sometimes when I hear people I do [identify mistakes] when it's the gender wrong or something like I pay attention to it, or when they conjugate wrong in future”. Participants also found mistakes after automated translations. For example, Participant 4 uses an online translator to understand messages and often must go back to the original message to understand the meaning, as the translator does not work correctly; Participant 10 also related to this: “If some things are not making sense, sometimes I can even tell what it thought I meant and why it translated it wrong”.

3.3. Algorithms

The Algorithms facet was related to about 20% of the statements. Participants employed systematic approaches for expressing themselves in Hebrew. They reported structuring their thoughts and ideas before writing them. Many follow a mental algorithm for conjugating verbs, considering various steps in a specific order. Participant 7 said to follow steps in conjugating, especially in his first years in Israel, by first recognizing the structure the verb belongs to and then applying the format. Participant 10 reported asking herself questions to know which steps to follow: "I guess the process is like, 'What tense do I need? Which person do I need to talk about?' And then formula”.

4. Discussion

The aim of the current study was to investigate how CT skills relate to strategies used while learning a second language, specifically in the case of new immigrants in Israel learning Hebrew. Of the six main facets described by Shute et al. (2017), the three most prominent were Abstraction, Debugging, and Algorithms; the facets that were only little related were Decomposition, Iteration, and Generalization. This points out to where CT is important in SLA, and highlights areas where the integration of CT into SLA could be further studied. By intentionally integrating CT into language teaching, educators and learners can benefit from a more effective learning environment, leading to language proficiency and higher sociocultural integration.

The Abstraction facet was the most prominent. This coincides with the idea supported by Aho (2012) that this facet is at the heart of CT. Participants heavily relied on their ability to analyze language data, identify patterns, and create mental models, to navigate the complexities of Hebrew as L2. We also note that Debugging is a crucial skill in L2 learning, which aligns with the importance of error analysis in language acquisition, as the mistakes learners identified enable the understanding of their mental grammar system (Gass & Selinker, 2008). In the field of CT, this finding connects to Aho’s (2012) idea that as systems become more intricate, often we discover that our model does not reach a solution, and one needs to research and create new models for the system. This is parallel to the process of mental grammar, as the learner is constantly acquiring more information and adjusting it to the system of the language being learned. Although less frequent, we found that the Algorithm facet was also employed during perception and expression of L2, particularly in important or complex cases. Learners apply systematic methods when processing and producing language to reach the ideal expression, in accordance with the study on students’ engagement with algorithms, as noted by Jacob and Warschauer (2018).

The CT framework used here (Shute et al., 2017) was suitable due to it being context-independent and comprehensive, thus allowing the study of problem-solving in the context of SLA; recently, it was used to study music learning (Regev Cohen et al., 2025), and its usefulness should be further studied.

References

Aho, A. V. (2012). Computation and computational thinking. *Computer Journal*, *55*(7), 833–835. https://doi.org/10.1093/comjnl/bxs074

Bounou, A., Lavidas, K., Komis, V., Papadakis, S., & Manoli, P. (2023). Correlation between High School Students' Computational Thinking and Their Performance in STEM and Language Courses. *Education Sciences*, *13*(11). https://doi.org/10.3390/educsci13111101

Ellis, R. (1999). *The Study of Second Language Acquisition* (2nd ed.). Shanghai Foreign Language Education Press.

Gass, S. M., & Selinker, L. (2008). *Second Language Acquisition: An Introductory Course* (3rd ed.).

Hsieh, H. F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research*, *15*(9), 1277–1288. https://doi.org/10.1177/1049732305276687

Hsu, T. C., Chang, C., Wu, L. K., & Looi, C. K. (2022). Effects of a Pair Programming Educational Robot-Based Approach on Students' Interdisciplinary Learning of Computational Thinking and Language Learning. *Frontiers in Psychology*, *13*. https://doi.org/10.3389/fpsyg.2022.888215

International Organization for Migration. (2024). *World Migration Report 2024*. United Nations Research Institute for Social Development.

Jacob, S. R., & Warschauer, M. (2018). Computational Thinking and Literacy. *Journal of Computer Science Integration*, *1*(1). https://doi.org/10.26716/jcsi.2018.01.1.1

Kanwal, R., Hussain, M. S., & Farid, A. (2022). Second Language Acquisition (SLA) and Bilingualism: Impact of Mother Tongue on English as a Second Language (ESL) Learners. *Research Journal of Social Sciences & Economics Review*, *3*(3). https://doi.org/10.36902/rjsser-vol3-iss3-2022(100-111)

Keating, G. D. (2016). *Second Language Acquisition: The basics*. Routledge.

Liang, Y. (2022). The application of mind mapping in English foreign language teaching: A case study of unit 1 great cities in Asia of the English textbook for grade six. *OALib*, *09*(03), 1–14. https://doi.org/10.4236/oalib.1108523

Regev Cohen, T., Armon, B., & Hershkovitz, A. (2025). The associations between computational thinking and learning to play musical instruments. *Education Sciences*, *15*(3), 306. https://doi.org/10.3390/educsci15030306

Sabitzer, B., Demarle-Meusel, H., & Jarnig, M. (2018). Computational thinking through modeling in language lessons. *IEEE Global Engineering Education Conference, EDUCON*, *2018-April*, 1913–1919. https://doi.org/10.1109/EDUCON.2018.8363469

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. In *Educational Research Review* (Vol. 22, pp. 142–158). Elsevier Ltd. https://doi.org/10.1016/j.edurev.2017.09.003

Yu, X., Soto-Varela, R., & Gutiérrez-García, M. Á. (2024). How to learn and teach a foreign language through computational thinking: Suggestions based on a systematic review. *Thinking Skills and Creativity*, *52*. https://doi.org/10.1016/j.tsc.2024.101517

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Young Children's Strategies for Developing Algorithmic Thinking in CT-Based Mathematical Problem-Solving Activities with Floor Robot

Ying ZHANG[1*], Gary WONG[2]
[1,2]The University of Hong Kong, Hong Kong SAR
zhangy15@connect.hku.hk, wongkwg@hku.hk

Abstract: *Although scholarly investigation of computational thinking (CT) has permeated into early childhood level over recent years, little is known regarding what early CT means from an interdisciplinary perspective and how it can be incorporated into early childhood settings through mathematics. To fill the gap, we integrate CT into six mathematical problem-solving lessons guided by embodiment design principles, and this study reports on findings regarding how algorithmic thinking, a foundational element of CT, is manifested in and supported by embodied cognition among 5- to 6-year-old children as they engaged in such activities with a programmable robot. Findings highlight different strategies young children adopted when developing algorithmic thinking, varying in levels of planning, predicting, and embodied interaction with the physical environment. Implications for integrating CT into early childhood education are discussed.*

Keywords: Computational thinking, Programming, Mathematics Education, Early childhood, Embodied cognition

1. Introduction

Computational thinking (CT) has gained wide recognition as an integral set of thinking skills applicable in problem-solving situations beyond computer science, necessitating everyone to grasp it from a young age to accommodate the increasingly computational world (Grover & Pea, 2013). Taking the domain general view of CT, researchers have explored the teaching and learning of CT by situating it into different subjects, with the aim of not only enriching our conceptual understanding of the interdisciplinary nature of CT but also informing steps to be taken for incorporating the essential thinking skills into K-12 education, such as in science, technology, engineering, mathematics, and the arts (Zhang et al., 2023). This integrated approach has also shown promise in alleviating the practical challenges if CT is taught through a newly standalone subject (Lee et al., 2020). Recently, studies on the CT integration into mathematics education have gained much insight into, for instance, the definition and operationalisation of CT in K-12 mathematical contexts (Weintrop et al., 2016), the complex interplay between the learning of CT and mathematics (Cui & Ng, 2021), and the cognitive mechanism between computational and mathematical thinking (Gerosa et al., 2021). However, the line of scholarly endeavour focusing on the early childhood level still remains obscure, considering young children's unique ways of thinking, learning, and expressing themselves among older students (Zeng et al., 2023). To fill the gap, this study set out to explore how early CT can be developed in programming-based mathematical contexts with a particular focus on number and arithmetic.

Algorithmic thinking (AT) is central—among all the other CT skills—when developing sequential and logical steps to be followed in solving a problem, and it has drawn increasing attention over the years (Clarke-Midura et al., 2021; Wong et al., 2024). According to a recent systematic review exploring the interplay between early CT and mathematics education (Zhang & Wong, 2024), AT is shown to be the predominant CT component intersecting widely with early mathematical concepts in numeracy, geometry, and measurement. For instance, when devising step-by-step solutions for a programmable robot moving toward a desired destination, young children need to determine whether the robot should move in the *direction* it is currently facing or rotate left or right, the *distance* from the current position to the next stop, as well as the *number* of steps needed in between (Berson et al., 2023; Welch et al., 2022). As such, the young CT learners are provided with opportunities to develop, exercise, and reinforce mathematical knowledge and skills throughout their CT engagement. In this study, we take AT as the foci of our attention and aim to shed light on what strategies young children employ in their development of AT. We posed the following research questions to guide this study: What are the AT strategies young children demonstrate when engaged in CT-based mathematical problem-solving activities with Bee-bot, and how do these strategies emerge during the process?

2. Method

As an initial stage of a larger project aimed at seamlessly integrating CT into early number and arithmetic learning with different programming tools (e.g., Beebot, ScratchJr), the first author conducted six 1-hour lessons with 20 children aged 5-6 in a Chinese kindergarten over six weeks during children's free play time. Participants were divided into 2 groups of 10 children, one using Beebot (an educational floor robot) and another with ScratchJr (a graphic programming application on iPad). The sessions were conducted with one pair of students each time, resulting in ten sessions each week. At each session, the pair of students was engaged in programming-based mathematical activities concerning number sense and basic arithmetic, such as counting, number comparison, addition, and subtraction. During the sessions, the researcher encouraged the pair of students to collaborate with each other and probed into their thinking process with questions, such as "How did you know that?" and "Why did you do that?". Data collection was conducted upon obtaining ethical approval from the authors' institution and informed consent from the participants' parents and the kindergarten principal.

The teaching sessions intentionally integrate four CT elements (i.e., algorithms, events, loops, and conditionals) into the six number and arithmetic lessons, and this present study reports on those sessions with Beebot. A map with 6 × 10 15*15 cm grids was prepared for children to program the Beebot to move. The lesson design follows Kotsopoulos et al.'s (2017) CT pedagogical framework consisting of (1) unplugged, (2) tinkering, (3) making, and (4) remixing, and we also employed the embodiment design principle (Abrahamson & Lindgren, 2022), especially in the first stage of *unplugged*, when introducing abstract CT elements to young learners at the first time. Pens and pencils were provided to children to support reasoning or express their thinking. All sessions were video recorded from different angles with several cameras.

The first author reviewed and transcribed the video data, including both verbal and nonverbal interactions (e.g., gestures and bodily movement) between the pair-learners and the researcher on a qualitative analysis software, MAXQDA. A multimodal discourse analysis was conducted to first capture representative episodes that characterise young children's AT demonstrated during the sessions and then draw connections among those episodes (Wortham & Reyes, 2020). Iterative refinement was performed through multiple rounds of data review, with codes adjusted to ensure alignment with emerging themes. The analysis is grounded in the embodied cognition perspective (e.g., Barsalou, 2008) to examine how young children's AT is manifested through and supported by their embodied actions. Children's drawings and the researcher's field notes served as supplements for triangulating the findings.

3. Findings

We identified two types of AT strategies, which will be elaborated on in this section: (1) *step-by-step execution* with immediate feedback from the Beebot's movement and (2) *Multiple-code execution* with delayed feedback. These strategies involve different levels of planning, predicting, and embodied interaction with the physical environment.

3.1. Step-by-step execution

In the initial phase of the lessons, where most of the children were introduced to the programmable robot for the first time, we observed their reliance on a *step-by-step execution* strategy. Specifically, children tended to input one command at a time, monitoring the robot's response before proceeding with the next instruction. This reflects a trial-and-error approach to algorithm construction, whereby children incrementally develop their algorithms by relying on immediate feedback from the Bee-bot to evaluate their current input and guide subsequent actions. This AT strategy is explorative in nature and may be attributed both to the ongoing development of young children's working memory and to the design features of the Bee-bot, which lacks an interface for visualising, tracking, or refining planned computational solutions.

3.2. Multiple-code execution

While novice learners initially adopted a step-by-step execution strategy for iterative AT development, a more sophisticated approach gradually emerged over the course of the interviews: children began programming the robot by inputting multiple commands at once before receiving feedback from the Bee-bot's actions. As they progressed toward this more complex strategy, we identified three distinct ways through which children supported their thinking process, each reflecting different levels of embodiment: (1) physically moving the robot step-by-step to simulate each command, (2) inputting code with one hand while tracing the expected movement on the mat with the other hand, (3) using pointing gestures in the air to simulate the movement, and (4) mentally simulating the movement without physical actions.

Inputting multiple commands before pressing the "Go" button on the robot requires children to operate with delayed feedback from the Bee-bot, where children were observed to leverage their whole body as cognitive resource in anticipating the Beebot's subsequent movements, either by manually relocating the robot or by using one hand to simulate how the robot will move on the grid. For the former, to illustrate, Yanni initially relied on a *step-by-step execution* strategy with immediate feedback during the first three sessions. However, in the fourth session, he spontaneously devised a new strategy: before inputting each command, he would first lift the Beebot and physically place it in a position which he intended the robot to move towards. Upon this embodied simulation, he would then press the directional button matching the movement he had just enacted. By repeating the *simulating-coding* process, Yanni was able to mitigate the cognitive load on his working memory arising from the Beebot's design feature. It is noteworthy that after observing Yanni's approach, his peer, Zack, adopted the same strategy in his subsequent programming practices. For the other form of embodied strategy in children's algorithm construction processes, for instance, Ken employed a reverse *coding-simulating* approach, where he input each command with one hand, followed by stretching his body to trace the Beebot's intended path on the grid with another hand. In addition to the highly kinaesthetic strategies, such as physically relocating the robot or manually predicting the routes, simulation was also completed by children's use of pointing gestures in the air without physical interaction with the robot, as demonstrated by Wesley. Furthermore, some children were able to input multiple commands before execution without relying on any embodied simulation, either spontaneously (e.g., Ken) or following encouragement from the researcher (e.g., Leo).

4. Discussion and Conclusion

This study investigates the strategies employed by young children, characterised by varying levels of embodiment, to support their development of algorithmic thinking during six-week CT-integrated mathematical lessons using a floor robot. The findings indicate that young children constructed algorithms through either step-by-step execution with immediate feedback from the robot (trial-and-error based) or multiple-code execution with delayed feedback involving more structured planning. In relation to the latter, four levels of embodied strategies were identified to facilitate their thinking process: physically relocating the robot, tracing the intended path with the hand, simulating the path through pointing gestures, and mentally envisioning the robot's movement. The emergence of these AT strategies suggests that early CT extends beyond abstract cognitive abilities requiring systematic and logical reasoning but is deeply rooted in children's embodied interaction with the environment, including the robot, peers, and the researcher. Future studies shall further investigate the role of embodied cognition in the development of other CT elements and identify effective pedagogical approaches to support this mode of thinking.

References

Abrahamson, D., & Lindgren, R. (2022). Embodiment and Embodied Design. In R. K. Sawyer (Ed.), *The Cambridge Handbook of the Learning Sciences* (3rd ed., pp. 301–320). Cambridge University Press.

Barsalou, L. W. (2008). Grounded Cognition. *Annual Review of Psychology*, *59*(1), 617–645.

Berson, I. R., Berson, M. J., McKinnon, C., Aradhya, D., Alyaeesh, M., Luo, W., & Shapiro, B. R. (2023). An exploration of robot programming as a foundation for spatial reasoning and computational thinking in preschoolers' guided play. *Early Childhood Research Quarterly*, *65*, 57–67. https://doi.org/10.1016/j.ecresq.2023.05.015

Clarke-Midura, J., Silvis, D., Shumway, J. F., Lee, V. R., & Kozlowski, J. S. (2021). Developing a kindergarten computational thinking assessment using evidence-centered design: The case of algorithmic thinking. *Computer Science Education*, *31*(2), 117–140. https://doi.org/10.1080/08993408.2021.1877988

Cui, Z., & Ng, O. L. (2021). The interplay between mathematical and computational thinking in primary school students' mathematical problem-solving within a programming environment. *Journal of Educational Computing Research*, *59*(5), 988–1012. https://doi.org/10.1177/0735633120979930

Gerosa, A., Koleszar, V., Tejera, G., Gómez-Sena, L., & Carboni, A. (2021). Cognitive abilities and computational thinking at age 5: Evidence for associations to sequencing and symbolic number comparison. *COMPUTERS AND EDUCATION OPEN*, *2*. https://doi.org/10.1016/j.caeo.2021.100043

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/ 0013189x12463051

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A Pedagogical Framework for Computational Thinking. *Digital Experiences in Mathematics Education*, *3*(2), 154–171.

Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational Thinking from a Disciplinary Perspective: Integrating Computational Thinking in K-12 Science, Technology, Engineering, and Mathematics Education. *Journal of Science Education and Technology*, *29*(1), 1–8. https://doi.org/10.1007/s10956-019-09803-w

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Welch, L., Shumway, J., Clarke-Midura, J., & Lee, V. (2022). Exploring Measurement through Coding: Children's Conceptions of a Dynamic Linear Unit with Robot Coding Toys. *EDUCATION SCIENCES*, *12*(2).

Wong, G. K. W., Jian, S., & Cheung, H.-Y. (2024). Engaging children in developing algorithmic thinking and debugging skills in primary schools: A mixed-methods multiple case study. *Education and Information Technologies*. https://doi.org/10.1007/s10639-024-12448-x

Wortham, S., & Reyes, A. (2020). *Discourse analysis beyond the speech event*. Routledge.

Zeng, Y., Yang, W., & Bautista, A. (2023). Computational thinking in early childhood education: Reviewing the literature and redeveloping the three-dimensional framework. *Educational Research Review*, *39*, 100520.

Zhang, Y., Ng, O.-L., & Leung, S. (2023). Researching Computational Thinking in Early Childhood STE(A)M Education Context: A Descriptive Review on the State of Research and Future Directions. *Journal for STEM Education Research*, *6*(3), 427–455. https://doi.org/10.1007/s41979-023-00097-7

Zhang, Y., & Wong, G. K. W. (2024). Exploring the Interplay of Computational Thinking and Mathematics in Early Childhood Education: A Systematic Review. *2024 IEEE Global Engineering Education Conference (EDUCON)*, 1–10. https://doi.org/10.1109/EDUCON60312.2024.10578653

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

A Study on Using Scratch Games to Deepen Students' Understanding of Sustainable Development

Wai Leung WONG
Ma On Shan Methodist Primary School, Hong Kong SAR
wlwong@mosmps.edu.hk

Abstract: *This study investigates the effectiveness of an environmental education game designed using Scratch. Through simulations of five geographical environments and real-time feedback mechanisms, sixth-grade students were guided to understand the relationship between human activities and environmental changes. The research adopted a blended learning approach, combining game-based experiences with classroom discussions across four thematic lessons. Results indicate that gamified learning effectively enhanced students' environmental awareness and learning motivation, promoting their understanding of sustainability issues.*

Keywords: Environmental Education, Gamified Learning, Scratch, Sustainable Development, Environmental Conservation

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

運用 Scratch 遊戲深化學生對可持續發展重要性的研究

黃偉樑
馬鞍山循道衛理小學，中國香港
wlwong@mosmps.edu.hk

【摘要】 本研究探討運用 Scratch 設計環境教育遊戲的教學成效。透過五種地理環境場景的模擬及即時反饋機制，引導小學六年級學生理解人類活動與環境變遷的關聯。研究採用混合教學模式，結合遊戲體驗與課堂討論，實施四節主題式課程。結果顯示遊戲化學習能有效提升學生的環境意識及學習動機，促進對永續發展議題的理解。

【關鍵字】 環境教育；遊戲化學習；Scratch；可持續發展；環境保育；

1. 引言

面對日益嚴峻的環境問題，傳統環境教育較難讓學生深入理解人類活動與環境的互動關係。本研究旨在探討運用 Scratch 教育遊戲如何提升小學生對可持續發展的認識及環保意識。具體研究目標包括：（1）探討遊戲化學習在環境教育中的成效；（2）評估學生在認知、情意及技能三個層面的學習成果；（3）分析混合式教學模式在環境教育中的應用價值。

為達成上述目標，本研究設計 Scratch 教育遊戲《地球守護者：生態平衡挑戰》，採用混合教學模式，探討遊戲化學習在環境教育中的應用成效。透過虛擬環境的互動體驗，引導學生分析人類活動對地球的影響，並思考可持續發展方案。

2. 研究方法

2.1. 研究設計

本研究採行動研究法，分為遊戲設計、教學實施及成效評估三階段。遊戲設計以 Scratch 開發《地球守護者：生態平衡挑戰》，包含五種地理環境場景（沙漠、高山、平原、森林、海洋）及四個環境變數（資源、人口、溫度、能源）的即時反饋系統。教學實施採混合教學模式，規劃四節主題式課程，運用遊戲體驗、小組討論及案例分析等策略。成效評估採多元評量工具，分析學生在認知、情意及技能的表現，並進行資料交叉驗證。

2.2. 研究對象與實施

本研究以六年級一個班級共 25 名學生為研究對象，實施為期四節課的教學活動，每節課 35 分鐘，總計 140 分鐘。研究採用多元方式收集資料，包括課堂觀察記錄、學生學習工作紙、遊戲互動數據、前後測問卷及學生作品分析。教師透過這些多樣化的資料來源，全面評估學生的學習成效及遊戲化教學的實施成果。本研究存在樣本規模較小、觀察時間有限等限制，且僅針對單一年級層進行研究，未能進行長期追蹤研究，這些限制可能影響研究結果的推論性，建議未來研究可擴大研究範圍，並進行更長期的追蹤觀察。

3. 實施框架

本研究規劃 140 分鐘課程，包含課前預習及三個教學階段。學生先透過 Scratch 進行遊戲體驗，觀察環境變數變化。第一節課（35 分鐘）引導認識地球資源；第二、三節（70 分鐘）分析環境問題，進行案例討論及方案設計；第四節（35 分鐘）學習可持續發展概念，規劃綠

色生活計畫。每個環節結合遊戲體驗、討論及實作活動，強調知識應用與實踐。教師擔任引導者，協助學生從遊戲經驗中建構環境知識，培養環保意識與行動力。

4. 教學反思與觀察

透過此次研究實踐，觀察到遊戲化學習確實能提升學生的環境教育學習成效。在課堂實施過程中，學生展現出高度的學習參與度，主動探索遊戲中的環境議題，並在小組討論時積極分享觀察與想法。《地球守護者：生態平衡挑戰》的即時反饋機制，有效幫助學生理解人類活動與環境變遷的複雜關聯。當學生在遊戲中做出決策後，能立即觀察到對環境造成的影響，這種直觀的因果關係展示，大幅提升了學生對抽象環境概念的理解程度。

然而，研究也發現部分學生需要教師額外引導，才能將遊戲中獲得的環境知識轉化為日常生活中的具體環保行動。這反映出環境教育除了知識傳遞外，更需要強化實踐層面的連結。混合式教學模式在本研究中展現出良好效果，透過結合數位遊戲與傳統課堂活動，既保持了學習的趣味性，又能透過面對面討論深化環境議題的思考。教師在課堂中適時引導學生反思遊戲經驗，有助於強化知識內化與實際應用。

基於研究觀察，建議未來在實施類似教學活動時，可進一步強化遊戲與實際生活的連結，設計更多具體的環保行動方案，並持續追蹤學生的環境行為改變。同時，教師在運用遊戲化教學時，需要注意適時調整教學策略，確保所有學生都能充分參與並從中獲益。

5. 結論

本研究顯示 Scratch 教育遊戲能有效支援環境教育，提升學生的學習興趣及環境議題思考深度。透過虛擬環境決策體驗，學生發展系統思維及問題解決能力。研究成果展現數位遊戲在環境教育的應用潛力，建議教育工作者運用遊戲化策略促進互動學習，持續優化教學設計，以培養具環保意識的公民。

參考文獻

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). *The Scratch programming language and environment. ACM Transactions on Computing Education*, 10(4), 16:1-16:15. https://doi.org/10.1145/1868358.1868363

Meya, J. N., & Eisenack, K. (2018). *Effectiveness of gaming for communicating and teaching climate change. Climatic Change*, 149(3), 319-333. https://doi.org/10.1007/s10584-018-2254-7

Ouariachi, T., Olvera-Lobo, M. D., & Gutiérrez-Pérez, J. (2017). *Gaming climate change: Assessing online climate change games targeting youth produced in Spanish. Procedia - Social and Behavioral Sciences*, 237, 1053-1060. https://doi.org/10.1016/j.sbspro.2017.02.154

Puttick, G., & Tucker-Raymond, E. (2018). *Building systems from scratch: An exploratory study of students learning about climate change. Journal of Science Education and Technology*, 27(4), 306-321. https://doi.org/10.1007/s10956-017-9725-x

Wu, J. S., & Lee, J. J. (2015). *Climate change games as tools for education and engagement. Nature Climate Change*, 5(5), 413-418. https://doi.org/10.1038/nclimate2566

Zhang, L., & Nouri, J. (2019). *A systematic review of learning computational thinking through Scratch in K-9. Computers & Education*, 141, 103607. https://doi.org/10.1016/j.compedu.2019.103607

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Enhancing Students' Computational Thinking and Creative Problem-Solving Skills Through AI Technology and Environmental Concepts: A Case Study of "AI Recycling Bin" Project for Grade 6 Students

Wai Lung MUNG[1], Chi Yan WONG[2*], Wai Lam CHU[3]
[1,2,3]Fung Kai No.1 Primary School, Hong Kong SAR
mungsir@fk1ps.edu.hk，astomwcy@fk1ps.edu.hk

Abstract: *This study examines the effectiveness of an Artificial Intelligence (AI) curriculum in enhancing primary school students' computational thinking abilities. The research, conducted with 52 sixth-grade students in Hong Kong, involved a 6-hour curriculum structured into four comprehensive units. Using the Six-step STEM methodology integrated with self-directed learning elements, the curriculum focused on three dimensions: knowledge of AI capabilities and limitations, development of ethical AI usage attitudes, and mastery of technical skills including HuskyLens applications and MakeCode programming. Through pre-test and post-test questionnaires, results revealed statistically significant improvements ($p<0.05$) in students' computational thinking abilities, particularly in problem decomposition skills, demonstrating the curriculum's effectiveness in developing both computational thinking and AI literacy.*

Keywords: AI Literacy, Self-Directed Learning, Six-step STEM Pedagogy, Computational Thinking, STEAM Education

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

結合 AI 技術與環保理念培養學生運算思維能力與創意解難能力：以六年級「人工智慧環保回收箱」課題為例

蒙韋綸 [1]，黃智仁 [2*]，朱偉林 [3]
[1,2,3] 鳳溪第一小學，中國香港
mungsir@fk1ps.edu.hk，astomwcy@fk1ps.edu.hk

【摘要】 本研究探討 AI 課程對 52 位於香港就讀的小學六年級學生運算思維及創意能力的影響。採用 STEM 六步曲結合自主學習四元素，課程包含 6 小時教學內容，著重知識(AI 概念及限制)、態度(AI 素養)及技能(HuskyLens 應用)三層面。前後測結果顯示學生運算思維有顯著改善($p<0.05$)，尤其問題拆解能力提升最明顯，支持 AI 教育結合自主學習能有效提升運算思維及 AI 素養。

【關鍵字】 人工智慧素養；自主學習；STEM 六步曲教學法；運算思維；STEAM 教育

1. 前言

隨著AI技術迅速發展，培養學生運算思維與AI素養愈發重要。現有研究多集中於中學階段，小學AI教育尤其結合自主學習的研究較少（Kong, 2021）。雖有研究指出程式設計活動可提升運算思維(Grover & Pea, 2013)，但AI與環保議題結合的效果研究不足。本研究為六年級學生設計校本課程，採用STEM六步曲（Kong, 2024）並融入自主學習元素（Knowles, 1975），重點包括開發小學AI課程、探討其對運算思維的影響及分析自主學習策略在AI教育的成效。

2. 課程設計理念

本課程立基於科技融入、問題導向學習及跨學科整合三大支柱，遵循"Learning by Doing"原則（Dewey, 1938），讓學生通過設計 AI 環保回收箱，探索技術應用價值。課程強調培養元認知能力（Flavell, 1979），鼓勵學生反思技術的倫理與社會影響，同時整合科學、技術、工程及數學知識，培養解決複雜問題的綜合素養（Yakman & Lee, 2012）。

3. 課程目標與學習成果

本課程通過「人工智慧環保回收箱」專題，培養學生系統化思考、演算法設計及方案評估能力（Wing, 2006）。預期學習成果涵蓋三個維度：知識層面理解 AI 基本概念與應用限制；技能層面掌握運算思維拆解問題與設計 AI 應用；態度層面發展 AI 素養及負責任科技使用觀念（Touretzky et al., 2019）。

4. 學與教策略

課程採用自主學習四元素融入 STEM 六步曲教學法（Kong, 2024）：「自學」階段學生獨立探索 AI 概念；「導學」環節由教師提供引導；「共學」階段分組協作解決問題；「互學」環節通過作品展示深化知識。同時強調探究式學習與動手實作（Kolb, 1984），使用多元形成性評估提供即時回饋（Black & Wiliam, 2009）。

5. 課程內容

學生使用 iPad 結合 MakeCode 平台進行積木式編程，開發 AI 環保回收箱。學習流程遵循設計思維（Brown, 2008）：從同理理解、定義挑戰、構思方案到製作測試。6 小時課程分四單元：AI 基礎認知(1 小時)、環保議題探討(1 小時)、HuskyLens 實作(2 小時)和成品製作(2 小時)。學生利用 HuskyLens 感測器設計分類算法，結合 micro:bit 與伺服馬達製作智能系統。針對班級差異(6A 理解力高，6B 動手能力強)，採分層任務：6A 著重算法優化，6B 加強基礎概念並提供額外支援。

6. 課程結果與分析

前測顯示約 60%學生對運算思維能力持正面評價，後測中問題拆解(+9.2%, p=0.03)、演算法設計 (+8.2%, p=0.04) 及知識遷移 (+8.2%, p=0.04) 提升最顯著，部分學生評估更加謹慎，反映對運算思維複雜性理解加深（Brennan & Resnick, 2012）。創意思維方面，「創意問題解決」提升 6.2% (p=0.05)，「創意作品產出」提高 4.4% (p=0.07)，班級差異 (6A:65%，6B:48%) 後期縮小 (6A:68%, 6B:59%)，顯示課程對起點較低學生效果更明顯（Craft, 2005）。相關分析顯示兩種能力呈中度正相關 (前測 r=0.58，後測 r=0.64)，特別是知識遷移與創意解難能力 (r=0.62)，運算思維高分組創意思維平均分 (4.2) 顯著高於低分組 (3.4, p=0.02)，證實能力互相促進（Mishra & Henriksen, 2018）。

7. 教學反思與建議

研究顯示結合 AI 與環保議題的課程能有效提升運算思維和創意解難能力。建議教師：強化思維策略遷移促進認知靈活性（Bransford et al., 2000）；實施差異化教學滿足不同學生需求（Tomlinson, 2014）；整合運算與創意思維訓練發揮協同效應；培養成長型思維建立正確能力認知（Dweck, 2006）。未來研究可設對照組，以准實驗設計進一步驗證課程成效。

參考文獻

Black, P., & Wiliam, D. (2009). Developing the theory of formative assessment. *Educational Assessment, Evaluation and Accountability*, *21*(1), 5-31.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*.

Dewey, J. (1938). *Experience and education*. Kappa Delta Pi.

Dweck, C. S. (2006). *Mindset: The new psychology of success. Random House*.

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, *42*(1), 38-43. https://doi.org/10.3102/0013189X12463051

Knowles, M. S. (1975). *Self-directed learning: A guide for learners and teachers*. Association Press.

Kong, S. C. (2024). Cross-disciplinary STEM learning for Asian primary students: Design, practices, and outcomes. In W. W. M. So, Z. H. Wan, & T. Luo (Eds.), *Cross-disciplinary STEM learning for Asian primary students* (pp. 147-163). Routledge.

Tomlinson, C. A. (2014). *The differentiated classroom: Responding to the needs of all learners* (2nd ed.). ASCD.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Enhancing Travel Graph Concept Learning with Computational Thinking

Tsz-Wai YUEN
Chinese Y.M.C.A. Primary School, Hong Kong SAR
ytw@cymcaps.edu.hk

Abstract: *This study explores the feasibility of integrating computational thinking into mathematics education through hands-on activities and programming simulations. By engaging in practical experiences, students learn mathematical graph concepts through a "learning by doing" approach. The study examines how programming a dice-rolling board game can help students understand key travel graph concepts such as direction, distance, and elapsed time. Through coding-based simulations and graphical representations, students transition from concrete activities to abstract travel graphs. This approach demonstrates the potential of incorporating computational thinking into mathematics lessons to enhance students' understanding of graphical data and mathematical reasoning. This research is significant as it highlights the value of computational thinking in fostering problem-solving skills and mathematical comprehension. By bridging hands-on activities with digital simulations, the study provides an innovative way to improve students' engagement and analytical skills, preparing them for future STEM learning.*

Keywords: Computational Thinking, mathematics education, Travel Graph

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

以運算思維概念支援行程圖概念學習

袁梓慧
中華基督教青年會小學，香港
ytw@cymcaps.edu.hk

【摘要】 本文研究透過由「造中學」的過程探索「運算思維」於數學科圖表類型學習的可行性。以程式玩棋子遊戲體驗「同向」和「相向」兩種主要類型的行程時間經過、方向和距離，利用程式模擬繪畫行程圖，由具體活動至抽象圖表認識行程圖的基本概念，分析將運算思維帶進數學課堂的可能性。本研究通過將實體活動與數碼工具結合，本研究提供了一種創新的學習方式，以提升學生的學習參與度和分析能力，為未來的 STEM 學習做好準備。

【關鍵字】 運算思維、數學教育、行程圖

1. 前言

教育界提倡運算思維在數學教育中幫助學生以邏輯推理和演算法思維解決問題。本研究以數學教育探索如何透過運算思維與數碼工具提升學習效能。通過實物操作與程式模擬的學習方式，從具體情境出發過渡至抽象概念，從而理解數學中的圖表和數據分析。研究將探討這教學模式的可行性並記錄其對數學學習的影響。

2. 運算思維

Wing（2006）提出運算思維（Computational thinking）為借用電腦思維模式用作解決困難、設計系列及剖析人類活動等。Pérez（2018）發現涉及幾何學的程序與以學生為中心的教學法，於數學推理與運算思維之間的循環互動有助產生數學知識。Barcelos 等（2018）分析運算思維於數學學習相關的文獻，發現涉及平面幾何、代數、符號轉換等認知數學思維有高相關性。吳藹藍等（2023）指出計算思維是涉及製定和解決問題的思維模式，它透過計算手段來表示和執行解決方案。

3. 造中學

Dewey（1986）提出了「做中學」（Learning by doing）的教學理念，主張學生透過情境引發出的問題，由主動動手測試和練習，反思和建構知識，從而獲得學習的印象和建構出知識。Papert（1991）於建造論（Construcionism）提及學習者對於事物的操作經驗作累積、分享和反思，從過程中發展出對事物的想法和瞭解。Papert 引領的「造中學」增加了個人化於學習上的重要性。學習者由模仿到創造的過程，加入個人的操控能發揮更多學習的可能性，從而深化學習及成為知識的擁有者，由高效自主性提升學習滿足感。

4. 行程圖

於二零二零年《小學數學課程闡釋》中提及「教師宜透過合適的學習活動，例如說故事，引導學生認識行程圖，以及透過闡釋行程圖，解涉及速率的應用題。」（香港教育局，2020）學生首次接觸以線性圖表方式記錄和表達兩個變數之間的關係，除了記錄行程之外，將來亦會用來記錄不同的實驗結果。

5. 教學設計理念

以 Scratch 棋子遊戲畫出線段，實踐 Kolb（1984）體驗式學習四階段：具體經驗（Concrete Experience）、反思（Reflection）、抽象概念化（Abstract Conceptualisation）、主動實驗（Active Experimentation）。學生擲骰子移動棋子，於模擬器記錄移動過程，觀察推測及分析線性圖的變化，驗證移動的方向和距離之間的關係。以同儕互動加強學習自主性、解難能力和創造性。讓學生了解行程圖的製作過程及針對方向的教學難點。

6. 教學實踐情況

活動以 Scratch 擲骰子和繪圖模擬器配合實際座位面向操作分為四個階段，第一階段處理擲出加數，認識橫軸和縱軸對應意思及前進和停留。大部份學生明白起點設於左下角，前進時線段會向上斜，停留時變平。能力較高的學生能表達擲出 2 比 1 大，即每回合行得較快，線段顯示較斜。第二階段處理擲出減數，學生大致能分辨移動方向。有學生提出疑惑為什麼向後退的速率較高，線段愈斜，應該向前行才有速率。此疑問是涉及方向的難點，因有實物助證，由學生帶出「-2」就是向後行兩步比向後行一步多，線段會較斜。學生成功用實際操作經驗解決了教學難點。第三階段二人並肩而坐向相同方向移動，學生發現兩個項目之間會有相距，有時會較近，即學生說的「差不多追到你了」。第四階段學生二人面對面坐反方向而行。學生能按自己的面向作出移動。學生能掌握何謂相遇，例如說出二人在同一點，行程圖上的線重疊，行程圖上出現交叉等。以工作紙作記錄和鞏固課堂的觀察和結果，讓學生認識數學詞彙以便回到課本的情景，將橫軸回合過渡至時間，縱軸棋子移動距離過渡為公里或米。

7. 反思和結語

學生透過探究式學習反覆嘗試和討論建構出行程圖的概念。學生探索會有不同觀察，教師宜於每一個階段緊扣教學重點作總結，確保學生能按進度認識教學點。如再設計小任務着較高能力學生善用模擬器深化學習會更好。活動以運算思維技巧來拆解行程圖，利用編程工具能節省課時，亦能提升學習興趣，繪製行程圖能補足課程中省略的部份。應用運算思維及編程工具於數學教學取得了成效，從課堂觀察中收穫了豐富的討論，相信運算思維於數學教育有更多的發展潛力與空間，值得教學工作者嘗試作出更多的研究。

參考文獻

Barcelos, T. S., Muñoz-Soto, R., Villarroel, R., Merino, E., & Silveira, I. F. (2018). Mathematics Learning through Computational Thinking Activities: A Systematic Literature Review. *J. Univers. Comput. Sci.*, *24*(7), 815-845.

Dewey, J. (1986). Experience and education. *The Educational Forum*, *50*, 241-252.

Kolb, D. A. (2014). *Experiential learning: Experience as the source of learning and development*. FT press.

Ng, O., Leung, A., Ye, H. (2023). Exploring Computational Thinking as a Boundary Object between Mathematics and Computer Programming for STEM Teaching and Learning. ZDM Mathematics Education.

Papert, S., & Harel, I. (1991). Situating constructionism. *constructionism*, *36*(2), 1-11.

Pérez, A. (2018). A framework for computational thinking dispositions in mathematics education. *Journal for Research in Mathematics Education*, *49*(4), 424-461.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

附錄：

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Exploring STEAM Education: A Curriculum on Clean Energy Powered Boats Integrating Scientific Experiments, Artificial Intelligence Teaching, and Computational Thinking Education

Wai Han CHEUK[1*], Kam Yuen LAW [2*], Lin Chun KWONG[3*]
[1,2,3]The Education University of Hong Kong Jockey Club Primary School, Hong Kong SAR
law_sir@hotmail.com, whcheuk@edujcps.edu.hk, lckwong@edujcps.edu.hk

Abstract: *STEAM education is a teaching approach that integrates five disciplines: Science, Technology, Engineering, Art, and Mathematics. It cultivates students' creativity, problem-solving skills, and teamwork spirit through interdisciplinary learning and practice. STEAM education has long been regarded as essential for nurturing students' innovative thinking and problem-solving abilities. This study aims to explore an innovative teaching curriculum that combines scientific experiments, artificial intelligence education, and computational thinking education, centered around the theme of clean energy powered boats. Through interdisciplinary learning and practice, the curriculum seeks to stimulate students' interest and motivation in the STEAM fields, further enhance their skills in studying scientific knowledge, and promote scientific literacy and innovation, thereby nurturing talent.*

Keywords: STEAM Education; Scientific Experiments; Artificial Intelligence Teaching; Clean Energy Powered Boats; Computational Thinking Education

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

探索 STEAM 教育：結合科學實驗、人工智能教學及運算思維教育的潔淨能源 動力船研習課

卓煒嫻[1*]，羅金源[2*]，鄺靈俊[3*]
[1,2,3]香港教育大學賽馬會小學, 香港
law_sir@hotmail.com, whcheuk@edujcps.edu.hk, lckwong@edujcps.edu.hk

【摘要】 STEAM 教育融合科學、技術、工程、藝術和數學，通過跨學科學習與實踐，培養學生的創造力、解難能力及合作精神。本研究以潔淨能源動力船為主題，結合科學實驗、人工智能及運算思維教育，激發學生對 STEAM 的興趣與學習動機，提升科學技能，推動創科學習，培育具創新力的人才。

【關鍵字】 STEAM 教育; 科學實驗; 人工智能教學; 潔淨能源動力船; 運算思維教育

1. 前言

STEAM 教育被視為培養學生創新思維和解難能力的關鍵（Chan et al., 2021）。本校自 2017 年起積極推動運算思維教育,成為香港賽馬會運算思維教育計劃的先導學校,並於 2020 年起連續三年成為五所資源學校之一。本文記述小五 STEAM 研習課主題「潔淨能源動力船」,學生運用運算思維結合科學實驗與人工智能學習，掌握科學與技術基礎，設計小組專屬「潔淨能源動力船」。課程透過探索與驗證，引導學生建構創新意識與實踐能力，為香港未來創科發展培育人才。

2. STEAM 研習課與跨領域統整教學實踐

香港特別行政區政府於《行政長官 2023 年施政報告》中提出，大力推動 STEAM 教育，包括在小學設立科學科，並於 2024/25 學年起在高小及初中推行編程教育和創科訓練，以培育本地創科人才，回應社會對科技創新和人才發展的需求（香港特別行政區政府，2023）。《小學教育課程指引》（2024）進一步強調「立德樹人重啟迪、創造空間育全人」的核心理念，更新七大學習宗旨，聚焦 STEAM 教育、媒體與資訊素養等能力，培育學生共通能力、探 究思維及主動學習精神，為未來發展奠基。

本校於 2023-2024 學年開設「STEAM 研習課」，以跨學科思維、媒體素養及創新精神為 核心，透過動手實踐及跨領域學習，激發學生創造力及解難能力。課程結合科學探究、設計 思考與創客精神，推動素養導向的學與教，同時建立教師 STEAM 教學檢核機制，確保教學 質素，全面促進學生在 STEAM 領域的綜合能力發展（王子華和蔡寶桂，2023）。

3. 「探索為根，驗證為本」的學習歷程

課程緊密結合聯合國可持續發展目標 7「經濟適用的清潔能源」，提供強調科學探究、設 計思考與實踐應用的學習體驗，旨在培養學生的跨學科思維、解難能力及對全球可持續發展 議題的關注（香港教育大學，2023）。學生以「如何利用潔淨能源為動力船提供動力？」為主 題，自選研究能源（如太陽能、風能、化學能等），探索其應用效能。學生運用生成式 AI 工 具蒐集資料，分析能源的優缺點及應用場景，並結合生成式圖像設計動力船模型圖，將構思 具象化。隨後，學生利用創客工具與運算思維能力（如 Micro:bit 編程，結合 3D 打印、發泡 膠切割或膠樽等材料）製作實物模型，實現設計與實踐的結合。完成模型後，學生

進行性能 測試，記錄及分析數據，對比不同潔淨能源（如太陽能板、風力發電機）的效能，參與小組比 賽並以數據驗證設計可行性。根據測試結果，學生反思設計不足，提出改進方案，深化其設 計思維與創新能力。

4. 人工智能在 STEAM 教學應用

在課程中，學生運用生成式 AI 進行資料搜集、分析與驗證，建構個人化筆記，並透過生 成式圖像將設計具象化。在明確目標下，學生與人工智能互動，進行深度思考，激發創造性 思維，幫助他們在數據洪流中精準捕捉有用資訊，蒐集具價值的網絡資料（Brown et al., 2020）。 過程中，學生需具備媒體素養，辨識資料真偽，篩選修正內容，通過辯證與驗證篩選出優質 資料，進行正確分類與組織，提升對資料的深層理解（Dwivedi et al., 2021）。

課程以學生為主體、教學目標為導向，鼓勵學生建構創造性學習成果。通過知識整合與模組化驗證，實現知識的可預測性與實踐性，全面提升認知能力（教育部，2023）。在 STEAM 研習課中，學生認識潔淨能源（如風能、太陽能、位能及化學能），進行實驗與測 試，並設計以潔淨能源推動的「Hong Kong Tour Boat」模型。船身繪有香港特色圖案，宣揚 可持續旅遊理念，回應《香港氣候行動藍圖 2050》中「淨零發電」與「綠色運輸」的減碳 策略，實踐「碳中和」。並將作品帶到「英國教育科技博覽會 2024」作國際交流分享，學 以致用，以創科能力驅動未來發展，解決現實生活的問題，推動社會發展。

課程以聯合國可持續發展目標（SDG）為主題，結合科學實驗、運算思維及人工智能的 STEAM 教學框架，透過實踐激發學生對科學與科技的興趣，並培養創造力與解難能力。人工智能的應用幫助學生有效分析資料，實現個性化學習，同時增強對可持續發展議題的關注及環保意識。STEAM 教育強調知識的整合與應用，讓學生靈活運用技術解決實際問題，為未來發展奠定基礎。課程不僅促進學生的全面發展，還推動社會的可持續發展，為香港創科人才的培養貢獻力量。

参考文獻

教育部. (2023). **人工智能在教育中的應用與挑戰**. https://www.moe.gov.cn

香港教育局. (2023). **小學科學教育課程發展**.

王子華和蔡寶桂. (2023). **從清華 STEAM 學校 DDMT 教學模式談 AI 融入 21 世紀跨領域統整課程之教學 實踐**. https://shorturl.at/gdO3U

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877-1901.

Chan, L. K., Wong, T. Y., & Lee, C. H. (2021). The impact of STEAM education on primary students' creativity and problem-solving skills. *Journal of Educational Research*, *45*(3), 123-135.

Dwivedi, Y. K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., ... & Williams, M. D. (2021). Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *International Journal of Information Management*, *57*, 101994.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Interdisciplinary Integration of "Light Properties" and Programming Instruction: A Teaching Practice for Cultivating Computational Thinking with Scratch

Hon Wai MOK[1*], Wing Ting YUEN[2*]
[1,2]Islamic Primary School, Hong Kong SAR
mohw@islamps.edu.hk, yuwt@islamps.edu.hk

Abstract: *This is an example illustrating a pedagogical practice for teaching the "Properties of Light" unit in General Studies through the pedagogical approach of "to play, to think, and to code" . Fifth-grade students utilize Scratch's sprite tools to simulate light rays projected onto virtual plane mirrors, observe the reflection direction of the light, and complete game-based tasks. To address the unique learning needs of non-Chinese-speaking students, the lessons consistently adopt a "learning by doing" approach. This hands-on methodology enables students to connect classroom knowledge through active experimentation, fostering both innovative thinking and computational thinking skills.*

Keywords: Computational Thinking, Scratch Programming, Learning by Doing, Interdisciplinary Teaching, Properties of Light

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

透過跨學科「光的特性」培養學生運算思維之教學實踐

莫漢威 [1*]，阮泳婷 [2*]
[1,2] 伊斯蘭學校，中國香港
mohw@islamps.edu.hk，yuwt@islamps.edu.hk

【摘要】 本文闡述一個透過「先遊戲、後思考、再編程」（to play, to think and to code ）的課堂實踐模式來教授常識科單元—光的特性。五年級學生利用 Scratch 中的角色工具，模擬光線投射到虛擬的平面鏡後，觀察光線的反射方向，並完成遊戲所要求的任務。為配合非華語學生學習的獨特性，課堂會經常以「邊幹邊學」（learning by doing） 的模式進行，讓學生在動手過程中連繫課堂知識，培養創新思維和運算思維能力。

【關鍵字】運算思維；Scratch 程式設計；邊幹邊學 ；跨學科教學設計；光的特性

1. 前言

運算思維作為 21 世紀核心素養，強調通過分解、抽象化、演算法設計等策略解決複雜問題（Wing, 2006）。近年研究指出，運算思維不應局限於電腦學科，而需與科學、數學等學科融合（Weintrop et al., 2016）。例如，科學現象模擬（光的反射理論）可轉化為程式設計任務，促進學生對科學規律的演算法表達。運算思維與跨科目的課堂結合，學生從學習經歷及實踐中找出錯處並作出修正，再延續發展及優化。

2. 編寫「光的特性」遊戲

五年級學生會在本單元建立一個模擬光線反射定律的程式，透過學習使用（forever）、(if else) 指令，測試當光線接觸鏡面圖示後所引發的其他程式碼。傳統教學依賴公式記憶與實驗觀察，但學生常難以理解動態過程。Scratch 的視覺化程式設計與物理引擎（如光線角度、座標移動）為模擬光行為提供了低門檻工具，學生可通過代碼“具象化”科學規律（Brennan & Resnick, 2012）。學生並不單單設計程式，更要運用「算法思維」（algorithmic thinking）透過編程來判斷入射角與鏡面所構成的角度，從而決定反射光線的方向。

3. 教學實踐過程

根據「邊幹邊學」理論，教師於課堂設計和教學法上需提倡“多動手、多思考”的教學活動來刺激學生的學習興趣。於施教上運用核心三步曲：先遊戲、後思考、再編程（to play, to think and to code），以助學生完成課堂的學習目標。

學生會先試玩完整版的遊戲，接著，學生需要思考程式的運作，為免學生盲目跟從老師指示堆砌出程式碼，老師使用試誤法（trial and error method）（Starch, 1910）的教學方法，鼓勵學生改變程式碼先後次序或參數（見圖 1），過程中進行實時測試，與組員討論、找出錯處並作出修正，學生在過程中能加深認識對運算思維中序列的概念。

透過以上活動，學生從探究中加深了解程式的運作及程序，這個過程可以深化他們對運算思維實踐中「測試及除錯」的理解。

課堂中老師鼓勵學生將程式優化，透過運算思維實踐中「重用及組合」，優化場境及程式碼，再設計另一個更高難度的關卡，過程中學生從運算思維實踐中經歷「反覆構思及漸進編程」反覆構思及漸進編程（見圖 2）。學生認為這樣的優化，能讓使用者有多一重思考的機會，加深對數碼思維的啟發及實踐，這樣更能適切地照顧初學者及不同學生的學習需要。

圖 1

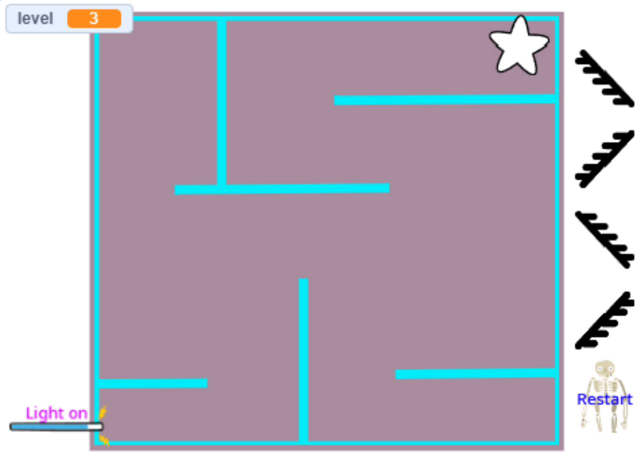


圖 2

4. 反思：運算思維對學生的發展需要

運算思維教育可幫助學生由單純的科技消費者培育為一群科技應用的創造者。學生創意是無限的，只欠一個合適的平台，賽馬會運算思維教育就正正提供一個平台，讓學生從日常生活中發揮數碼創意，並為他們對創新科技的發展和應對未來的挑戰作好準備。教授運算思維不單單只讓學生學習編寫程式，更重要是課堂上容讓學生出錯，這點正正與一般學科科目的課堂剛剛相反，一般學科都是由老師指出錯處，但教授運算思維卻是鼓勵學生嘗試、出錯，讓他們自行找出錯處並作出修正，這正正有助發展學生自主學習的能力，培養他們運用數碼科技解決問題之高階思維。

5. 結論

透過跨學科的教學實踐，學生在學習光的特性和反射定律的同時，能夠有效地培養其運算思維與編程能力。Scratch 作為一個創意編程平台，不僅能激發學生的興趣，還提供了一個實踐抽象科學概念的有效途徑。通過將“光的特性”與 Scratch 程式設計結合，學生不僅掌握了科學知識，更在演算法設計中內化了分解、抽象與邏輯推理能力。其次，運算思維中「測試及除錯」和「反覆構思及漸進編程」對小學階段亦極為重要，學生從失敗中學習，作出修正，繼而再進一步構思及優化，這正是現今學生生活於人工智能年代必須具備的重要元素。

參考文獻

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *AERA 2012*.

Kong, S. C. (2016). A Framework of Curriculum Design for Computational Thinking Development in K-12 Education. *Journal of Computers in Education*, *3*(4), 377-394.

Starch, D. (1910). A Demonstration of the Trial and Error Method of Learning. *Psychological Bulletin*, *7*(1), 20.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology*, *25*, 127-147.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Interdisciplinary Project-Based Learning Curriculum Design in Information Technology—A Case Study of "Chasing Light, Youth"

Xi ZHANG[1], Yiying ZHANG[2*]
[1]Beijing Normal University,China/Beijing
[2*]Longteng School of Longhua District, China/Shenzhen
870553925@qq.com,zhangyy51@qq.com

Abstract: *Against the backdrop of the Fourth Industrial Revolution, the rapid development of technologies such as artificial intelligence (AI) has raised new demands for interdisciplinary talent cultivation in the education system. This paper takes the "Chasing Light, Youth" project as an example to explore the design of an interdisciplinary information technology project based on the "learning progression" concept. Through a three-phase curriculum—"Light-Chasing Voyage," "Light-Chasing Growth," and "Light-Chasing Transcendence"—systematic and coherent learning modules replace traditional fragmented activities, integrating on-campus and off-campus resources with multidisciplinary knowledge to cultivate students' scientific literacy, innovative capabilities, and practical skills. This design aims to provide a curricular pathway for future educational reform.*

Keywords: Interdisciplinary learning, Learning progression design, Project-based learning, Information technology

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

信息科技跨学科项目学程设计——以“追光吧，少年”为例

张茜[1]，张艺莹[2*]
[1]北京师范大学，中国北京
[2*]龙华区龙腾学校，中国深圳
870553925@qq.com，zhangyy51@qq.com

【摘要】 在第四次工业革命背景下，人工智能等技术的快速发展对教育体系提出了跨学科人才培养的新要求。本文以“追光吧，少年”项目为例，探讨基于“学程”理念的信息科技跨学科项目设计。通过“追光启航”“追光成长”“追光超越”三阶段学程，以系统性、连贯性的学习模块替代传统碎片化活动，整合校内外资源与多学科知识，培养学生的科学素养、创新能力及实践能力。本设计旨在为未来教育模式改革提供学程化实践路径。

【关键词】 跨学科学习；学程设计；项目式学习；信息科技

1. 前言

科技的飞速发展推动了教育体系的变革。《新版义务教育课程标准》强调跨学科学习的重要性，要求通过项目式学习（PBL）强化学生的综合素养。然而，当前科技创新教育存在碎片化、简单化等问题，亟需以“学程”为核心的系统化设计。学程（Learning Program）不同于传统课程或单一活动，其以学习者为中心，围绕特定主题规划连贯的学习模块，强调知识整合与实践迁移（波斯纳，2019）。本文以“追光吧，少年”项目为例，构建以“光”为核心主题的跨学科学程，通过结构化设计突破学科壁垒，探索教育形态的全面超越。

“追光”具有双重隐喻：一是象征青少年对理想与科技的追求；二是体现科技创新对人类未来的引领作用。项目以学程化为核心理念，通过“体验—创造—迁移”的逻辑主线，实现从知识掌握到创新应用的递进，最终达成学科核心素养目标。

2. 学程设计的理论基础与框架

2.1. 学程的理论内涵

学程设计源于对传统课程与活动局限性的反思。根据波斯纳的定义，学程是“围绕两类特定主体（普及课与特色课学生）形成的结构化学习框架”（波斯纳，2003），其核心特征包括：系统性：模块间逻辑紧密，形成螺旋式上升的学习路径；整合性：融合多学科知识与真实问题解决；学生主体性：以任务驱动激发自主学习与协作能力。

2.2. 学程设计框架

项目分为三阶段，体现学程的递进性与整合性：追光启航（基础学程）：通过 3D 打印、激光切割等技术体验，激发学生兴趣，建立跨学科认知基础；追光成长（深度学程）：以未来社区设计为任务，融合编程、AI 创作等活动，培养问题解决与团队协作能力；追光超越（迁移学程）：整合校外资源（如无人机配送、机器人设计），实现知识迁移与创新应用。

2.3. 实施路径的双主线设计

教师主线：情境创设—任务规划—实施指导—成果评价，注重学程的系统化设计与资源整合；学生主线：发现问题—分析解决—成果展示—反思评价，强调学程对学生实践能力与创新思维的培养。

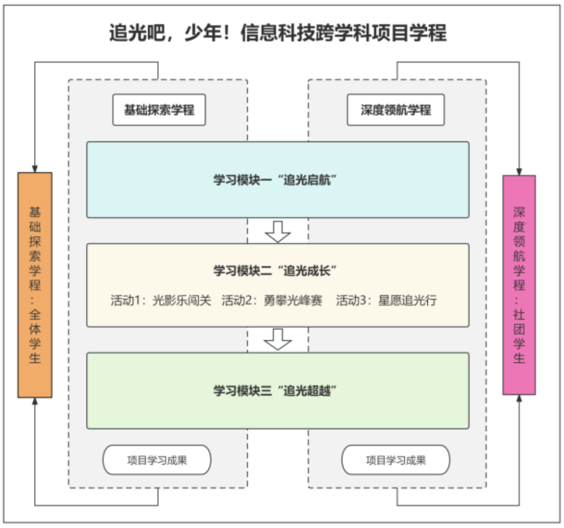


图 1 学程设计与实施逻辑

3. 学程化设计的创新与实践价值

3.1. 对传统教育的超越

系统性替代碎片化：以连贯的学程模块替代孤立活动，解决传统科技教育“装饰性开展”问题；跨学科整合：围绕“未来社区设计”等复杂问题，融合信息科技、艺术、工程等学科，形成整合性知识网络。

3.2. 学习方式的革新

“体验—创造—迁移”逻辑：突破传统课堂的线性讲授，通过校内外结合的实践任务（如科技公司研学），强化深度学习与创新应用；双主线协同：教师与学生的互动路径设计，保障学程的灵活性与目标导向性。

3.3. 情感与文化的融合

装置艺术载体：如“未来之光”互动装置，通过人脸识别、声光互动等技术，将科技与人文结合，激发学生情感共鸣；文化传承与创新：从传统皮影到 VR 体验，展现人类“逐光”历程，实现科技与传统文化的共生。

4. 作者资料

“追光吧，少年”项目通过学程化设计，实现了跨学科教育的多维度突破。其核心价值在于以系统性学程替代碎片化活动，通过真实情境中的实践与创新，培养学生科技素养与人文情怀。未来可进一步拓展学程设计的应用场景，如引入生成式人工智能辅助个性化学习，或通过校企合作深化校外研学模块，为基础教育提供更具普适性的学程范式。

参考文献

教育部 (2022)。 **新版义务教育课程标准**。
波斯纳, G. (2003)。 **课程设计：模块化与学程化路径**。

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Pedagogical Reflections on Computational Thinking: Using Programming and Computational Thinking to Enhance Students' Cross-Border Learning Effectiveness

Man Sing HSU[1]*, Mau Fai WONG[2]
[1,2] King's College Old Boys' Association Primary School No. 2, Hong Kong SAR
mshsu@kcobaps2.edu.hk，mfwong@kcobaps2.edu.hk

Abstract: *This paper illustrates how 6 primary students applied computational thinking to design an application called "Cross-Border Learning Diary" to address the shortcomings of traditional paper-based learning booklets used in cross-border learning activities. Through interviews and needs analysis, the students identified essential features for the application, such as multimedia recording, AI-powered real-time Q&A and translation, and interactive educational games, aiming to enhance the efficiency and experience of cross-border learning. During the design process, the students incorporated core computational thinking principles and underwent iterative testing and optimization, including simplifying operational workflows and improving the user interface design to address issues raised by test users. This project not only enabled the students to grasp the fundamental concepts of computational thinking but also fostered their digital creativity, problem-solving skills, and collaborative spirit. The paper highlights the value of computational thinking education - empowering students to grow through practice, collaboration, and innovation, and laying the foundation for them to become digitally skilled problem-solvers.*

Keywords: Primary Student, Computational thinking, Cross-Border Learning, Artificial Intelligence (AI), Digital Creativity

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

運算思維教育的教學實踐反思：

運用編程結合運算思維提升學生的跨境學習效能

許文星 [1]*，黃茂輝 [2]
[1,2] 英皇書院同學會小學第二校，香港
mshsu@kcobaps2.edu.hk ，mfwong@kcobaps2.edu.hk

【摘要】本文展示了 6 名小學生運用運算思維設計「跨境祖國學習日誌」應用程式，解決以往紙本學習冊在跨境學習中的不便與不足。他們透過訪談和需求分析，確立應用程式須具備多媒體記錄、人工智能(AI)即時提問與翻譯、趣味互動遊戲等功能，以提升跨境學習效能與體驗。在設計過程中，學生融合運算思維，反覆測試並優化操作流程和介面，有效解決用戶回饋的各個問題。這個項目讓學生不僅掌握了運算思維的核心概念，也培養了數碼創意、解難能力和協作精神。本文展現運算思維教育的價值-讓學生在實踐中成長，為成為具備數碼技能的解難人才打下堅實基礎。

【關鍵字】小學生；運算思維；跨境學習；人工智能；數碼創意

1. 前言

運算思維，也稱計算思維，指的是「運用計算機科學的基礎概念去解決問題、設計系統和理解人類的行為。」(Wing，2006)。香港教育局近年在高小年級推行運算思維和編程教育，其目的並非訓練及培養電腦程序編寫員，而是讓學生得到實作經驗及建立解難的信心，透過協作及重覆的測試來解決問題(教育局，2020)。由此可見，運算思維的發展著重的並非編程理論和技術，而是希望學生能透過適切計劃的學習活動，經歷和體驗「定義問題，解決問題，系統設計，以及透過電腦科學的基本概念理解人類行為(Wing，2006)」， 從而發展和成長為具備數碼創意的解難人才。

2. 運算思維教育的教學實踐

2.1. 定義問題 (problem identification)

運算思維和編程是解決問題的方法，可以應用在不同學習階段及不同學科(教育局，2020)，核心重點是引導學生多在日常生活中發掘問題或在數碼世界中提出問題(Kong，2022)。教師可鼓勵學生在參與不同的學習活動時多觀察、提問和討論，並思考利用數碼創意優化或提升學習活動體驗的可能性。

本校每年都最少會舉辦一次到祖國內地的跨境學習活動，藉活動提升學生對祖國歷史、文化的認識，以及與內地姊妹學校學生建立跨境友誼。然而，透過反思過往參與活動的經驗，學生認為在跨境活動中派發的紙本學習冊可以改善及優化，包括：學習冊無法即時紀錄相片或影片、學習冊以紙筆記錄，十分不方便、學習冊內的資料，缺乏互動、缺乏趣味等。

2.2. 解決問題 (problem solving)

透過綜合分析過往經驗以及訪談資料，6 名學生決定組成研發小組，並找了老師作為指導顧問，希望把他們學到的運算思維和編程知識學以致用。他們計劃透過 MIT App Inventor 2 研發一個應用程式，以取締學校的紙本跨境學習冊。研發小組認為應用程式應具備更豐富的功能，以滿足跨境學習時不同的使用需求，從而充分發揮電子工具的優勢。

2.3. 系統設計 (system design)

研發小組將應用程式命名為「跨境祖國學習日誌」，並設計了多項功能，以實現目標並解決所發現的問題。這些功能包括：

2.3.1. 我的跨境回憶與日誌

讓同學能使用文字或語音輸入，即時記錄跨境學習活動的見聞。此外，同學還可以通過表情符號表達當下心情，並以照片和影片保存回憶。多媒體記錄不僅比傳統紙筆記錄更方便，還更生動和更具吸引力。

2.3.2. 人工智能領隊

人工智能領隊主要提供兩項功能：第一是即時提問，通過 MIT App Inventor 2 的對話機器人(ChatBot)元件，讓同學在旅遊景點中能主動向 AI 提問，隨時獲取當地的歷史與文化資訊。第二是即時翻譯，幫助有需要的同學解決語言上的障礙，利用應用程式與祖國同胞進行簡單的交流與溝通。

2.3.3. VR 遊覽國家博物館

應用程式充分善用外部實用資源，通過網絡瀏覽器(WebViewer)元件連接至中國國家博物館的電子展廳，讓同學可以在正式參觀相關國家博物館之前，進行模擬遊覽和預習。

2.3.4. 我的跨境好友、好友聊天室

讓同學記錄在跨境學習中結識的新朋友，便於回港後保持聯絡。應用程式還設有好友聊天室，讓同學能直接通過應用程式與朋友交流、聯絡，提升對應用程式的使用黏著度。

2.3.5. 認識祖國遊戲區、跨境學習相片集

透過趣味互動的小遊戲，「認識祖國遊戲區」讓同學能輕鬆學習並深入了解祖國的地理、文化和歷史等資訊。而「跨境學習相片集」則讓同學可以透過相片記錄跨境學習旅程中的精彩瞬間，方便回港後自我回顧或與他人分享。這些具有高度互動性且充滿吸引力的功能，旨在提升應用程式的實用性，吸引用戶持續使用，並增強用戶對應用程式的黏著度。

3. 反思

在設計「跨境祖國學習日誌」應用程式過程中，學生充分體驗了運算思維的核心理念，包括定義問題、解決問題、系統設計及測試改進。他們從實際學習情景和需求出發，發現跨境學習中紙筆記錄的痛點，從而為應用程式設計了多個功能。在測試中，用戶反映應用程式部份功能操作過於複雜，且某些功能按鈕的設置不夠直觀，研發小組隨即透過反覆測試與優化，包括改善介面設計、簡化操作流程，並加入清晰的指引提示等作出改良。

4. 總結

本文探討了運算思維教育的重要性及其在小學階段的實踐價值與應用。本文案例展示了一群學生設計應用程式以促進跨境學習，幫助同學更有效地學習祖國歷史和文化。整個過程不僅提升了學生的數碼創意，還培養了他們積極面對挑戰的態度，充分體現了運算思維教育的價值。這種教育模式強調過程重於結果，通過協作與創新，為學生成為具備數碼技能的解難人才奠定了基礎。

參考文獻

教育局(2020)。**計算思維－編程教育：小學課程補充文件 (修訂版)** 。中華人民共和國香港特別行政區教育局。

Kong, S. C. (2022). Problem formulation in computational thinking development for nurturing creative problem solvers inprimary School. *Education and Information Technologies,* 1-20.

Wing, J. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Presenting Coding Through Problem Solving and Logical Thinking Model Implementing Coding Education in KS1

Yan-Wai-Wind HO *, Yuen-Ching YUNG
Tsuen Wan Catholic Primary School, Hong Kong SAR
hoyw@twcps.edu.hk, yungyc@twcps.edu.hk

Abstract: *While programming education has been widely implemented in primary schools, significant pedagogical challenges persist in introducing coding education to KS1. This study proposes an interdisciplinary pedagogical framework integrating mathematics and coding education, aiming to concrete abstract programming concepts through problem solving skills. Focusing on the "four-digit number" unit in primary 2, we designed a gamified instructional module utilizing Scratch-based visual programming platforms and developed systematic learning worksheets grounded in problem solving. The project aims to lead students to understand computational thinking through problem solving. This research contributes both practically through replicable curriculum prototypes and theoretically by proposing an "Early-stage Programming Literacy Development Framework," emphasizing the necessity of concept concretization and disciplinary anchoring for young learners. The findings provide empirical evidence for cross-domain instructional design and suggest new pathways for cultivating computational thinking in lower primary education.*

Keywords: problem solving, computational thinking, coding education, logical thinking, Key Stage 1

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

將編程以解難及邏輯思維模式呈現 於初小推行編程教育

何欣蔚*，翁琬晴
荃灣天主教小學，香港
hoyw@twcps.edu.hk，yungyc@twcps.edu.hk

【摘要】現時各小學均已經開始推行編程教育，惟較難於初小推行編程教育及讓初小學生理解運算思維。因此，本文目標是期望透過結合數學和編程教育，將編程及運算思維以解難及邏輯思維問題呈現，讓初小學生透過解難來理解編程，從而為學生於初小階段奠定編程能力的基礎。而本文按上述理念，以二年級的四位數為示例，製作出相對應的教案，包括 Scratch 製作的遊戲及對應的數學及編程學習工作紙，期望學生於學習過程中能夠以解難策略來理解運算思維。本文期望能夠啟發讀者，讓讀者能夠以其他形式來呈現編程及運算思維，發展初小學生的編程運算思維能力。

【關鍵字】 數學解難；運算思維；編程教育；邏輯思維；初小

1. 前言

現時，香港小學課程中，編程教育常在高小推行，旨在讓學生理解運算思維的概念（算法、抽象化、和自動化），再加以實踐，利用運算思維解決現實生活中的問題，培養學生拆解問題、處理數據、開發程序等高層次思維能力，成爲工具創造者（吳韶康、王瑞壎、陳致澄和柳賢，2024；教育局，2020；Wing，2006)。誠然，要求初小學生成爲工具創造者略顯困難，但學校亦能於各學科滲透運算思維，讓他們先作爲工具使用者的角色，體驗以模塊編程工具所製作的遊戲，再輔以教師深入淺出的講解，讓初小學生對運算思維的概念有初步的認識（教育局，2020）。此次結合運算思維的學科教學單元設計，我校以初小學生為對象，將編程教育融入數學科，抽取編程化為數學解難題目或邏輯問題，讓學生以解難及邏輯思考模式認識運算思維的基本概念，本文將闡述此教學設計。

2. 教學單元設計簡介

是次教學單元設計將運算思維結合數學科，選取了二年級課程中的數範疇（2N1）四位數課題，教師採用 Scratch 製作遊戲「齊來學四位數」，遊戲分爲「認識四位數」、「奇數和偶數」及「比較大小」三個部分，作為教材及鞏固學生所學的工具。此外，教師抽取了遊戲中的部分編程，將其化為數學解難及邏輯思維問題，製成分爲數學及編程兩部分的工作紙，讓學生在體驗遊戲後，由老師引導、在完成工作紙的過程中一步步瞭解此遊戲是如何透過編程實現的，從中認識運算思維的部分基本概念：算法（分支、序列、循環）及自動化。

3. 教學程序

3.1. 程序一 學生體驗 Scratch 遊戲內「認識四位數」的部分，以此鞏固四位數的各項知識點，包括各數位上數字代表的數值及四位數的讀法。然後，教師引導學生完成工作紙上的數學解難題目：學生需先以已有知識思考讀出某三位數 ABC 時，是否只有「A 百 B 十 C」一種讀法，從而發現當十位或個位的數字為「0」時，讀法會有所變化，然後進一步推論出四位數亦然。當百位、十位或個位的數字為「0」時，便會出現「A 千 B 百 C 十 D」以外的讀法。承接上題，接著學生需以數學解難方式（窮盡法）分別列出「0」和「1」可組成的三位及四位數組合，然後思考這些數的讀法。教師則會展示及講解編程內對應各種組合讀法的編程。

3.2. 程序二 學生體驗 Scratch 遊戲內「奇數和偶數」的部分，以此鞏固對奇數和偶數的判斷力，然後完成工作紙的數學部分並得出兩個結論：若某四位數的個位是 1/3/5/7/9，那麼該數便是奇數；若個位是 2/4/6/8/0，那麼該數便是偶數。接著，教師引導學生完成工作紙上的編程部分，認識「奇數和偶數」中核對答案的編程,當中包括「事件」和「控制-條件」。其中的邏輯為「當」點按奇數時，「如果」題目數字的個位是 1/3/5/7/9，「那麼」便正確;「如果」是 2/4/6/8/0，「那麼」便錯誤。讓學生從數學中認識編程內的邏輯思維。

3.3. 程序三 學生體驗 Scratch 遊戲內「比較大小」的部分，鞏固比較四位數數值大小的能力，然後完成工作紙的數學部分、歸納結論：比較兩個數的大小時，應先比較千位，若千位數字相同則比較百位，如此類推。接著，教師引導學生認識「比較大小」中核對答案的編程，並完成工作紙的編程部分。其中編程為「當」點按上面的數時，該數需符合某些條件，例如「如果」千位較大或千位相同但百位較大等，「那麼」便正確，若不符合條件便錯誤。

4. 預期結果

4.1. 數學教育 以 Scratch 遊戲「齊來學四位數」及工作紙的數學部分作爲教材，預期能鞏固學生對四位數各知識點的掌握，包括四位數的位值和讀法、奇數和偶數的區分以及比較大小的判斷。此外，教學流程及工作紙的設計亦能提升學生使用數學解難策略（窮盡法）解決問題（三位及四位數的組合）的能力，在歸納總結（判斷奇數和偶數及比較數字大小的方法）時亦能訓練學生以數學語言表達意見的能力。

4.2. 編程教育 預期學生能通過遊戲體驗及教師的講解認識運算思維的基本概念（算法及自動化）。例如：在進行遊戲時，當綠色旗被點擊，便會播放「齊來學四位數」的錄音。這些動作的先後次序所形成的「序列」便是建立算法時的基本編程結構之一。而在 Scratch 中，序列編程的實現便是透過「事件」積木。另一方面，教師引導學生完成工作紙上編程部分時，亦預期學生能進一步認識運算思維的基本概念。在認識「奇數和偶數」及「比較大小」部分的核對答案編程時，教師引導學生以「如果、那麼」的邏輯思維模式思考，讓學生認識建立算法的另一基本編程結構——分支：透過設定規則控制結果，若出現符合條件的動作便會帶來相對的結果，這是透過「控制」積木實現。最後，教師會補充在建立算法後，程式便可在不需人手的情況下不斷執行指令,讓學生知道這就是運算思維的另一概念——自動化。

5. 總結

運算思維和編程是一種可以在各個學習階段、各個學科中使用的解難及邏輯思考模式（劉吉軒，2018)。雖然初小學生較難理解算法及開發程序，但學校亦可將編程教育以初小學生也能理解的數字化、邏輯化方式融入數學科課程中，訓練學生的解難及邏輯思維能力，讓學生作爲工具使用者的角色先體驗以 Scratch 製作的遊戲，再輔以深入淺出的工作紙及講解，讓學生掌握運算思維的基本概念，對以模塊編程工具製作遊戲的方法有初步的瞭解，並開拓運算思維視野、連培養數碼充權能力，以作編程教育的啓蒙，奠下基礎從而銜接高小課程。

参考文獻

吳韶康、王瑞壎、陳致澄和柳賢（2024）。不插電運算思維學習活動開發之模式建構與示例。**臺灣教育評論月刊**，**13**（4），頁 135-144。

教育局（2020）。**計算思維—編程教育 小學課程補充文件**。香港教育局，頁 1-14。

劉吉軒（2018)。計算思維與數位人文研究意涵。**數位典藏與數位人文**，**1**，51–77。

Wing, J. M. (2006) Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

The Application of Computational Skills to Grant Proposal Writing

Yin Ling CHEUNG
Nanyang Technological University, Singapore
yinling.cheung@nie.edu.sg

Abstract: *Computational thinking skills are crucial to academic success. The bulk of the literature produced so far has done little in informing us how computational thinking skills impact academic writing activities, particularly occluded genre such as grant proposals. The identified research gap needs to be addressed because the application of computational thinking skills to academic writing is important in helping us understand how computational thinking skills impact thinking processes and therefore the quality of writing. This proposed study explores the integration of computational thinking into grant proposal writing within higher education, targeting graduate students and novice researchers who aspire to get teaching development grants. In my presentation, using my own award-winning proposal as an example, I will delineate a step-by-step approach adopting the core elements of computational thinking, including decomposition, generalization, abstraction, algorithmic thinking, and evaluation. These principles can be effectively applied to enhance the grant writing process. The findings may contribute to the broader discourse on computational thinking in academic writing in higher education. Ultimately, this research seeks to bridge the gap in the literature concerning the application of computational thinking in grant writing, providing actional insights for researchers and enhancing learning outcomes in the university setting.*

Keywords: computational thinking, writing, research proposal, higher education

1. Introduction

Computational thinking is defined as the ability to "solve problems, develop systems, and comprehend human behavior by applying the core principles of computer science" (Wing, 2006). Computational thinking has been applied in many fields including academic writing, which involves important problem-solving elements. Selby and Woollard (2013) suggested that five important components are embedded in computational thinking, namely decomposition, generalization, abstraction, algorithm thinking, and evaluation. Decomposition refers to a person's ability to break down problems. Generalization is about one's ability to apply a solution to comparable problems. Abstraction refers to one's ability to see the big picture. Algorithm thinking is about one's propensity to solve problems in a step-by-step way. Evaluation is about one's ability to analyze problems and solutions. In what follows, I will use my own research proposal titled "Integrating automated feedback and goal-setting to improve speaking skills in a graduate course" that was recently awarded funding as an example. I will describe and explain how computational thinking was applied during the grant proposal writing process. This is an important gap in the literature because few studies examined the application of computational thinking to grant proposal writing, which is an occluded genre.

2. Theoretical Framework

The activity theory that Leont'ev (1981) proposed is clear to show how individuals act to achieve the motive in different conditions. However, ironically, despite his emphasis on the social relation of individuals, it is ambiguous to show the relationship between individual subject and his/her social community (Engeström, 1999). The history that the subject had, the rules and roles that the subject are engaged, the community that they are affiliated are not visible (Swain, Kinnear, & Steinman, 2015). Thus, Engeström (1987) made an attempt to visually express relationship between individuals and social that Leont'ev had failed. Vygotsky (1978) and Leont'ev (1981) illustrated triadic representation of actions with subject, object and mediating artefacts. Engeström (1987, 1999) suggested a complex model of an activity system to consider not only the three main components that Vygotsky (1978) and Leont'ev (1981) suggested, but also rules, community and division of labour. In my proposed study, the subject refers to teachers and students. The object refers to the grant proposal (final product). Mediating artefacts refer to the components of computational thinking skills. The community is about the university environment. Rules are about the correct application of computational thinking skills to proposal writing. Division of labour refers to students writing the proposals, and reviewers giving the feedback on the proposals.

3. Methodology

The study will be conducted at a university in Singapore. Master and PhD students will be recruited. These students will provide written consent for their participation in the study. Data will not be collected without written consent. In terms of measurement, first, I will adopt a writing strategy survey developed by Petric and Czarl (2003). The survey contains 38 items testing the conscious actions and participants' behaviours in improving their writing. Eight items are related to pre-writing, 14 items are related to writing, and 16 items relating to revision of writing. Second, the computational thinking scale (Tsai et al, 2021) will be administered. The scale comprises 19 items measuring 5 constructs: decomposition (3 questions), abstraction (4 questions), algorithmic thinking (4 questions), evaluation (4 questions) and generalisation (4 questions).

4. Implications for Future Research and Pedagogical Practice

In my presentation, I will illustrate how computational thinking skills can be integrated to proposal writing. One important implication is that students and early career researchers will understand that computational thinking can encourage them to become more cognizant in the language, structure, and organization of a grant proposal. They should break down difficult problems to smaller parts, and then address them while writing a proposal.

References

Engeström, Y. (1987). *Learning by expanding: An activity-theoretical approach to developmental research.* Orienta Konsultit.

Engeström, Y. (1999). Activity theory and individual and social transformation. In Y. Engeström, R. Miettinen & R. Punamäki (Eds.), *Perspectives on activity theory.* Cambridge University Press.

Leont'ev, A. N. (1981). The problem of activity in psychology. In J. V. Wertsch (Ed.), *The concept of activity in soviet psychology.* M. E. Sharpe.

Petric, B., & Czarl, B. (2003). Validating a writing strategy questionnaire. *System, 31*(2), 187-215.

Selby, C. C., & Woollard, J. (2013). Computational thinking. The developing definition. In proceedings of the 45th ACM Technical Symposium on computer science education.

Swain, M., & Kinnear, P., & Steinman, L. (2015). *Sociocultural Theory in Second Language Education: An Introduction through Narratives* (2nd ed.). *Multilingual Matters.*

Tsai, M.-J., Liang, J.-C. Lee, S. W.-Y., & Hsu, C.-Y. (2021). Structural validation for the developmental model of computational thinking. *Journal of Educational Computing Research*, 1-18.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

The Cultivation of STEM Literacy for the Secondary Students with a Stent Bridge Model Project

On-chi SIU
Christian Alliance College, Hong Kong SAR
soc@cactm.edu.hk

Abstract: *A class of secondary students experienced the whole learning process in STEM through design and make (E) their own wooden stent bridge models with the concepts of structural mechanics, material sciences (S) and skills of scaffolding (T). They then tested (M) the bridges with loadings and searched for improvement ideas with own observations, and the discussion with teachers as well as their fellow classmates. The ideas would then be used and verified with the reconstruction of paper- tube stent bridges. Students would self-reflect the effectiveness of the whole STEM learning process with the self-evaluation questionnaire in STEM learning process to facilitate their applications in the other learning scopes and continuous to cultivate their STEM literacies.*

Keywords: STEM, STEM literacy, Self-directed learning, Learning by doing

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

中學生的科技創新素養培養：以「支架橋」模型專題為例

蕭安智
香港宣道中學，香港
soc@cactm.edu.hk

【摘要】一班中學生藉自行設計和製作「支架橋」模型，整全地經驗一次 STEM 學習模式，他們應用了與「支架橋」相關的結構力學、材料科學等概念(S)，以「支架搭建」技術(T)動手造「木支架橋」模型(E)，然後測試它的負重能力(M)，再藉觀察、與老師和同學探討測試結果，找出改良點子應用於搭建紙管「支架橋」模型(E)上，並再作測試(M)以證明點子的可行性。學生完成整個學習過程後，以《STEM 學習成效自評問卷》自我檢視和反思運用這模式的學習成效，以助他們日後應用於其他範疇學習，並持續他們的「科創素養」培養。

【關鍵字】STEM；科技創新素養；自主學習；手學

1. 前言

筆者藉「支架橋」模型專題，讓學生透過「動手做」自主研習相關 STEM 課題，並整全地經驗 STEM 相關學習技巧，盼望有助於他們長遠地培養出「科技創新素養」。當這個專題學習完結後，學生需填寫了一份《STEM 學習成效自評問卷》；筆者就著問卷調查結果分析和反思，寫成本文跟教育界同寅分享，期盼引發更多的討論；相信這些前線學與教的討論，於現時推行香港的 STEM 教育仍甚缺少。

2. 科技創新素養

科技創新素養(又稱：創科素養)目標是培養學生(聯合國教科文組織，2019；引自香港科技創新教育聯盟，2021)：(一) 俱備將不同學科概念、理論、原則聯繫並應用於生活解難的基本創科知識；(二) 掌握信息處理、問題解決、工程思維、科學探究、計算思維、創意與創新能力，又能安全正確使用科學技術設備，與及獨立工作和協作溝通的能力；(三) 對人和事擁有好奇心、誠信、客觀、開放性思想、勤奮、毅力、系統性、合作精神、責任心、嚴謹、冒險精神、道德化、決定和感恩等正確價值觀。學者們皆認為「科創素養」的培養，以「動手做」的實作學習與及以學生主導的「自主學習」這兩種模式為最佳 (林坤誼，2021)。

3. 「支架橋」模型專題的課堂設計

筆者是次課堂採用了 TMS 模式(New York State Education Department, 1997；引自林坤誼，2021)，即在單一「設計與科技」科中進行關聯式的統整，以「支架橋」模型為專題，透過不同學習活動，讓學生學習與結構力學、材料科學的相關概念(S)，並藉「支架搭建」技術(T)，先動手造「木支架橋」模型(E)，然後測試它的負重能力(M)，再藉觀察、與老師和同學探討測試結果，將從中獲得結論和構思(T)，再次應用搭建於紙管「支架橋」 (E)上，以測試(M)改良的點子是否可行，這令同學更深刻認識結構力學、材料科學的相關科學原理(S) (見圖片集)。當學生完成整個專題學習後，他們需要填寫《STEM 學習成效自評問卷》，問卷的設計參考了聯合國教科文組織(2019)的「21 世紀的科創素養框架」(引自香港科技創新教育聯盟，2021)分為三部份，包括：科創知識學習與掌握、科創的認知技能與運用與及科創態度和價值觀等。筆者相信，學生藉問卷調查的自我檢視，他們會更深刻認識這種「STEM」學習模式，有助他們日後在其他範疇的學習應用，並持續他們的「科創素養」培訓。

4. 結果與分析

是次，共有 47 名我校中三學生完成這個專題研習。按問卷調查結果(見附件)，有八成多學生對於自主「動手做」的實作學習模式感到非常喜歡，因為學生可不斷在過程中探索和反思，直至完成他們心目中的「支架橋」(Dewey, 1938, 1977；引自朱耀明，2011)，也能從中學習和應用了一些與「支架橋」的結構、材料製作相關的科學知識。當然，礙於課時限制，他們製作技巧和經驗又不足，竟有兩成多人認為自己不能一絲不苟地製作；這卻反過來說明他們的認真態度。在科創素養培訓方面，這次專題讓學生一次過整全地體驗 STEM 的各個學習進程。在過程中，學生(接近九成的回應)對學習事物充滿好奇心，敢於為他們心目中「承重最多、最穩固的支架橋」不斷嘗試，實屬難能可貴的科創素質。此外，有百分九十學生表示自己具備工程思維認知技能計算思維和科學探究技能力，但他們明顯探究的經驗和知識不足，影響了他們不懂怎樣從測試結果中找到需改善之處。九成多學生們似乎喜歡透過這種「動手做」的實作學習模式，去掌握科學與科技的知識。然而，他們仍舊慣於聆聽教師的指導，多於嘗試與老師和同學作有質素的學術討論，因學生表示喜歡討論的百分比跌回至八成。此外，他們亦甚少從網上自行學習。所以，筆者認為，要讓學生們長遠地培養出他們的「科創素養」，非單單一次的「動手做」專題體驗能辦到，仍是要根本地改變學生的思考和自學模式。

5. 結語

這次以「支架橋」模型專題的 STEM 模式學習，令學生投入於動手做、探索和思考，發揮了他們的學習的自發性、創意和解決問題的能力，並從中對相關的科學知識多了理解，卻突顯了他們的批判能力、分析思維、STEM 學習的經驗和背景知識之不足，以致他們未能從測試結果中找到可改善之處。因此，學生宜提早在初中甚或高小時，加強訓練他們以這種 STEM 模式學習。老師與學生們藉觀摩、討論和探索來學習的氛圍，都必須在課堂中鼓吹。此外，設計與科技科原以「動手做」實作學習為本，擁有優勢在課堂中推行更多 STEM 模式學習。在專題選取方面，宜與科學科溝通，將科技科的實作學習專題成果，也帶進科學科的課堂，讓學生進一步了解背後的科學概念。還有，數學科亦需要加強學生對數據閱讀、分析的訓練，以增強他們的批判思維及分析能力。

參考文獻

朱耀明 (2011)。「動手做」的學習意涵分析—杜威的經驗學習觀點。**生活科技教育月刊，2011 年，四十四卷，第二期**。

高寶玉(2018)。香港自主學習的探索：融合東西方理念的嘗試。**課程研究，13 卷 1 期**，2018 年 3 月，頁 29-53。

高寶玉 (2020)。香港 STEM 教育的挑戰及國際經驗的啟示。**香港 STEM 教育：理論與課堂實踐** (頁 6-17)，香港，香港教育大學卓越教學發展中心。

香港科技創新教育聯盟 (2021)。**「融入灣區・裝備自我」青年科創技能政策研究報告**。嶺南大學 STEAM 教育及研究中心。

林坤誼 (2021)。STEM 跨領域教學合作設計模式與有效失敗經驗設計之研究。**人文社會科學研究：教育類，第 15 卷，第 2 期**，頁 1-18。

附件 A

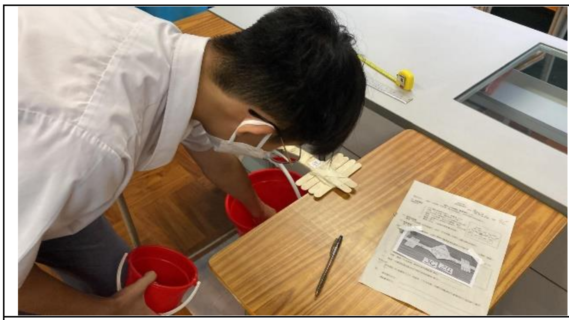


圖 1) 學生測試自行設計支架橋

圖 2) 學生測試自行設計支架橋

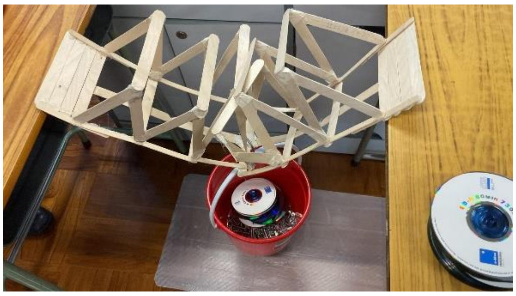


圖 3) 木支架橋的載荷測試

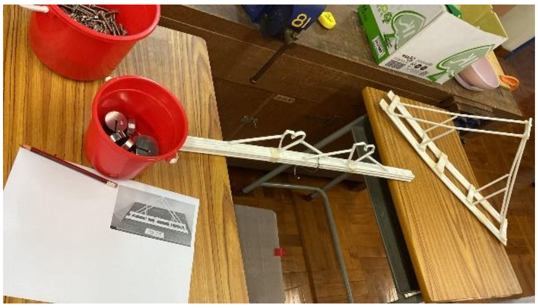


圖 4)紙管支架橋的載荷測試

圖 5)橋支架因負重變形，學生可見施力的走向

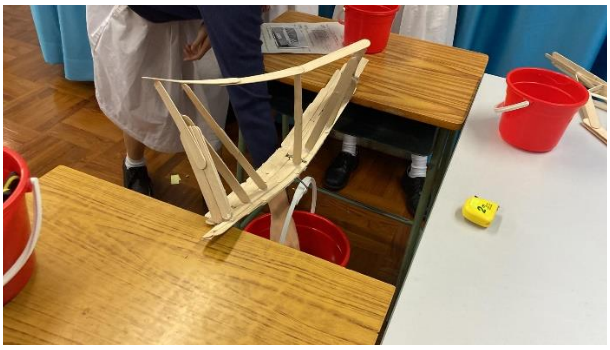


圖 6)橋支架因負重變形，學生可見施力的走向

附件 B

主題：「支架橋」模型專題 – STEM 學習成效自評問卷 (暨回應統計結果)

本問卷旨在讓學生完成整個「支架橋」模型專題後，自我檢視在學習與「支架橋」模型設計和製作相關的 STEM 知識和技能的成效，並「動手學」(Learning by Doing) 的學習模式的看法和態度。同學請反思整個學習過程，並誠實作答。謝謝！

科創知識學習與掌握

科創知識指如何將每個科創獨立學科的概念、理論、原則等聯繫的知識，並能解決問題所涉及的相關知識，例如：測量數據、保證信度和效度等；也包括跟具體領域或職業相關的技術性知識。

	(✓ 出你認為合適的選擇。)	十分不同意	不同意	同意	十分同意	無回答
1)	你能有效運用從網上搜尋與「木支架橋」模型設計的相關知識。(解決問題相關知識)	0.00%	10.64%	74.47%	14.89%	0.00%
2)	你能有效掌握與「支架橋」模型(包括：木和紙管) 相關的結構、力學等科學概念。(學科的概念、理論、原則等聯繫)	0.00%	8.51%	78.72%	12.77%	0.00%
3)	你能有效運用與「支架橋」模型相關的結構設計理論、材料特性的原則於「支架橋」製作。(學科的概念、理論、原則等聯繫) (解決問題相關知識)	0.00%	12.77%	72.34%	14.89%	0.00%
4)	你能有效藉網上筆記閱讀並完成相關習作，掌握與「支架橋」模型相關的科學與科技知識。(學科的概念、理論、原則等聯繫)	2.13%	12.77%	70.21%	14.89%	0.00%
5)	你能有效從老師指導中，掌握與「支架橋」模型相關的科學與科技知識。(學科的概念、理論、原則等聯繫)	2.13%	2.13%	74.47%	21.28%	0.00%
6)	你能有效從「支架橋」模型製作和測試過程中，掌握相關的科學與科技知識。(學科的概念、理論、原則等聯繫) (測量數據知識)	0.00%	8.51%	74.47%	17.02%	0.00%
7)	你能有效藉網上筆記閱讀，找到新點子應用於「支架橋」模型的設計和製作。(解決問題相關知識)	0.00%	8.51%	63.83%	27.66%	0.00%

科創的認知技能與運用

科創技能則包含三方面：1) 信息處理、問題解決、工程思維、科學探究、計算思維、創意與創新能力等認知技能；2) 正確和安全使用科學技術設備、儀器和標本等技術性技能；3) 既能獨立工作又能與他人協作的溝通技能

	(✓ 出你認為合適的選擇。)	十分不同意	不同意	同意	十分同意	無回答
8)	你能有效按自己的構思將「支架橋」模型(包括：木和紙管) 完整製成。(問題解決、工程思維的認知技能)(獨立工作能力)	0.00%	8.51%	61.70%	29.79%	0.00%
9)	你能有效運用製作工具和材料將「支架橋」模型(包括：木和紙管) 完整製成。(問題解決、工程思維的認知技能) (獨立工作能力)(正確和安全使用科學技術設備)	0.00%	8.51%	65.96%	25.53%	0.00%
10)	你能有效運用量度工具完成「支架橋」模型(包括：木和紙管)測試。(科學探究、計算思維的認知技能) (獨立工作能力) (正確和安全使用科學技術設備)	4.26%	2.13%	74.47%	19.15%	0.00%
11)	你能有效從「支架橋」模型(包括：木和紙管)的測試中，找出需要改善的地方。(科學探究、計算思維的認知技能)	2.13%	14.89%	65.96%	17.02%	0.00%
12)	你能有效從「木支架橋」模型的測試結果，找出新點子應用於「紙管橋」的設計和製作。(創意與創新能力的認知技能)	0.00%	8.51%	65.96%	23.40%	2.13%
13)	你能有效從老師指導中，找出新點子應用於「支架橋」模型(包括：木和紙管)的設計和製作。(創意與創新能力的認知技能) (與他人協作的溝通技能)	0.00%	8.51%	72.34%	19.15%	0.00%
14)	你能有效觀察、與同學討論，找出新點子應用於「支架橋」模型(包括：木和紙管) 的設計和製作。(創意與創新能力的認知技能) (與他人協作的溝通技能)	0.00%	10.64%	59.57%	29.79%	0.00%

<u>科創態度和價值觀</u>

科創態度和價值觀指：好奇心、誠信、客觀、開放性思想、勤奮、毅力、系統性、合作精神、責任心、嚴謹、冒險精神、道德化、決定和感恩等

	、	十分不同意	不同意	同意	十分同意	無回答
15)	你喜歡自行設計和製作「支架橋」模型(包括：木和紙管)，以達到承重最多、最穩固的要求。(好奇心) (冒險精神)	0.00%	10.64%	70.21%	19.15%	0.00%

	、	十分不同意	不同意	同意	十分同意	無回答
16)	你喜歡將「支架橋」模型(包括：木和紙管)進行測試，以證明自己的構思可行性。(好奇心) (冒險精神)	0.00%	17.02%	61.70%	21.28%	0.00%
17)	你喜歡從「支架橋」模型 (包括：木和紙管)的測試結果，找出「支架橋」模型可改善地方。(客觀)	0.00%	8.51%	72.34%	19.15%	0.00%
18)	你是一絲不苟地製作「支架橋」模型(包括：木和紙管)。(責任心、嚴謹)	0.00%	23.40%	59.57%	17.02%	0.00%
19)	你喜歡與老師和同學討論，以找出新點子應用於「支架橋」模型(包括：木和紙管)的設計和製作。(開放性思想、合作精神)	0.00%	19.15%	57.45%	23.40%	0.00%
20)	你會不斷改善你的「支架橋」模型(包括：木和紙管)，以達到承重最多、最穩固的要求。(嚴謹、勤奮、毅力)	0.00%	10.64%	59.57%	27.66%	2.13%
21)	你喜歡透過自行設計和製作「支架橋」模型，學習相關的科學與科技知識。(好奇心)	2.13%	14.89%	63.83%	17.02%	2.13%
22)	你喜歡與老師和同學討論「支架橋」模型設計，學習相關的科學與科技知識。(開放性思想、合作精神)	2.13%	17.02%	61.70%	19.15%	0.00%

(參考資料：聯合國教科文組織(2019)：《探索 21 世紀的科創素養》報告，引自香港科技創新教育聯盟，2021。)

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

The Feasibility of Using Artificial Intelligence to Explore Ecological Balance in Primary Education

Wai Leung WONG
Ma On Shan Methodist Primary School, Hong Kong SAR
wlwong@mosmps.edu.hk

Abstract: *This article presents research cases from Ma On Shan Methodist Primary School on integrating artificial intelligence tools into science education to study ecological balance. The research demonstrates that machine learning and generative AI can effectively deepen students' understanding of animal classification and ecosystems, increase engagement and learning outcomes, while developing digital literacy and critical thinking. However, implementation challenges include differences in student abilities, technical limitations, and time constraints. This study provides important reference for educators seeking to incorporate AI into primary school science education.*

Keywords: Artificial Intelligence in Education, Primary Science Education, Ecological Balance, Machine Learning, Generative Artificial Intelligence (GenAI)

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

在小學教育中運用人工智能探索生態平衡的可行性研究

黃偉樑
馬鞍山循道衛理小學，中國香港
wlwong@mosmps.edu.hk

【摘要】 本文介紹馬鞍山循道衛理小學將人工智能工具融入科學教育以學習生態平衡的研究案例。研究顯示，機器學習和生成式 AI 能有效加深學生對動物分類和生態系統的理解，提高參與度和學習成果，同時培養數位素養和批判思維。然而實施時需應對學生能力差異、技術限制和時間約束等挑戰。本研究為欲將 AI 整合入小學科學教育的教育工作者提供重要參考。

【關鍵字】 教育與人工智能應用；小學科學教育；生態平衡；機器學習；生成式人工智能

1. 引言

本研究探討人工智能工具在小學科學教育中的應用潛力，特別聚焦於生態概念的教學。通過馬鞍山循道衛理小學的兩個案例研究，分別運用機器學習進行動物分類教學，以及使用生成式人工智能模擬食物鏈的變化。研究顯示，人工智能工具不僅能使抽象概念具體化，提高學生參與度，還能培養其科技能力。然而，實施過程中需要考慮學生準備程度、技術設施和教學效能等因素。本研究為教育工作者提供實用的實施建議和最佳實踐方案。

2. 研究方法

本研究採用質性案例研究方法，透過課堂觀察、學生作品分析、Padlet 互動記錄、教師反思及學生意見等多元管道收集資料，探討人工智能技術在小學科學教育中的應用效果。研究包含兩個以六年級學生為對象的案例，每個案例均為 70 分鐘的教學單元。

第一個案例聚焦於運用機器學習技術進行動物分類教學，採用 Google Teachable Machine 作為主要工具，配合 Padlet 進行互動和作業提交，並使用 YouTube 提供學習資源。此設計旨在加強學生對動物分類特徵的理解和識別能力。

第二個案例探討生成式人工智能在食物鏈教學中的應用，使用特製的生態系統模擬器，讓學生觀察和預測人類活動對食物鏈的影響。教學過程結合 Slido 進行即時互動，使用 Padlet 收集反思，並通過 YouTube 提供教學視頻。

3. 實施框架

本研究的兩個案例採用三階段教學設計：課前準備、課堂活動和課後延伸，每個階段均配合適當的電子工具，確保學習的連貫性和深度。

案例一中，課前階段通過 YouTube 提供預習影片，學生在 Padlet 上記錄筆記和想法，建立基礎知識架構。70 分鐘的課堂教學主要進行引導式探索，教師帶領學生使用不同的電子教材進行動物分類實踐，並用 Padlet 記錄討論成果。課後則使用 Google Teachable Machine 進行延伸練習和評估活動。

案例二保持相似架構，但聚焦於生態系統探索。課前同樣使用 YouTube 和 Padlet 進行預習，課堂中學生使用生成式人工智能平台模擬生態系統變化，配合工作紙記錄觀察。課後通過 Slido 進行互動式複習和學習反思。

這種三階段框架的設計理念是：通過課前自主學習建立基礎認知，課堂上借助電子工具深化理解，最後通過延伸活動鞏固知識。雖然強調自主學習，教師仍需提供適當指導，確保教學效果。整個框架中的數位工具選用經過審慎考慮，以有效支援教學目標的達成。

4. 結果與分析

本研究通過多方面觀察和數據收集，發現人工智能工具在小學生態教育中成效顯著。學生對生態系統的理解，特別是食物鏈和生態平衡等複雜概念的掌握明顯提升。使用機器學習工具進行動物分類活動，不僅培養了學生的分類技能，還增強了其觀察和分析能力。在技術應用方面，Google Teachable Machine 的平均準確率超過 85%，有效提升教學品質和學習興趣。生成式人工智能模擬器則幫助學生直觀理解生態系統中的因果關係，通過互動式操作預測環境變化的影響，顯著提高了課堂參與度。然而，研究過程也面臨挑戰，包括網絡穩定性和設備可用性等技術基礎設施問題，以及教師備課時間投入大和學生能力差異等實務困難。這些問題雖然影響教學進度，但多能通過靈活調整和適當支援得到解決。

5. 討論

研究顯示，成功實施人工智能輔助教學需要多個關鍵要素配合。首先，結構化學習方法透過清晰的學習目標和漸進式難度調整，確保學習效果。當教師能靈活運用多元化學習途徑時，學生學習效果最為理想。其次，互動學習環境的營造促進知識交流和團隊協作，培養學生的溝通能力。即時回饋機制幫助學生調整學習策略，而實踐實驗則提供親身體驗和驗證概念的機會。完整的支援系統同樣重要，包括詳細的教師指引、技術支援和差異化學習材料。教師指引有助掌握教學節奏，技術支援確保課堂順利進行，而差異化材料則能照顧不同程度學生的需求，確保所有學生獲得適當的學習機會。

6. 建議

根據研究發現，建議學校採用循序漸進方式實施人工智能教育，先選擇部分班級或科目試點，再根據效果擴大規模。這種方法讓學校有充足時間評估和調整策略，同時讓師生逐步適應新的教學模式。教師專業發展方面，建議學校定期舉辦技術和教學法培訓工作坊，幫助教師掌握必要技能並有效融入教學。建立教師交流平台分享經驗和解決方案，也是重要的支援措施。資源分配方面，學校需要確保充足的基礎設施，包括硬件設備、網絡穩定性和學習材料。同時建立長期的技術支援機制，確保教學活動順利進行。這些措施對於成功實施人工智能教育至關重要。

7. 結論

本研究證實人工智能工具能有效提升小學生態教育效果，關鍵在於將技術應用與教學目標達致平衡。研究發現，人工智能的互動性和即時回饋特性顯著提升了學生的參與度，而生成式人工智能模擬器則幫助學生更好地理解生態系統的複雜關係。此外，教學設計的結構化和教師的引導角色對學習成效有重要影響。然而，技術基礎設施的限制和教師準備時間的投入仍是需要克服的挑戰。建議進行更多長期研究，評估人工智能教育的持續影響，並探索在其他學科的應用可能性。未來發展方向應包括評估長期學習效果、擴展跨學科應用，以及優化評估方法，為學生提供更優質的學習體驗。

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

The Fish Ball Game: Using Game-Based Constructivism to Address Lower Elementary Students' Confusion Between Multiplicand and Multiplier in Mathematics

Ting Hin, CHAN*, Ka Wai, LIU*
Fung Kai Liu Yun Sum Memorial School, Hong Kong SAR
kugitch@hotmail.com, liukw@fklys.edu.hk

Abstract: *This study explores how a game designed based on constructivist learning theory can support Primary Two students in understanding multiplication concepts, particularly in distinguishing the structural roles of the multiplicand and the multiplier. Grounded in the principles of digital game-based learning, the study involved a digital teaching experiment using The Fish Ball Game conducted in a Hong Kong primary school classroom. Employing a qualitative case study approach, the research revealed that many students harbored misconceptions about the structure of multiplication, and that their conceptual understanding did not always align with their academic performance. Through scaffolded support, immediate feedback, and peer interaction within the game, students were able to revise their existing cognitive schemas and achieve conceptual change. The findings underscore the significant value of digital tools in constructivist mathematics education, particularly in addressing persistent structural misconceptions.*

Keywords: Constructivist Learning, Scaffolding, Digital Game-Based Learning (DGBL), Multiplication Structure

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

魚蛋遊戲：運用遊戲式建構主義釐清初小學生對乘數與被乘數的混淆

陳廷軒*，廖嘉偉*
鳳溪廖潤琛紀念學校, 香港
kugitch@hotmail.com liukw@fklys.edu.hk

【摘要】本研究探討如何運用基於建構主義理論設計的遊戲，促進小學二年級學生對乘法概念的理解，特別是釐清「乘數」與「被乘數」的結構性區分。研究以數位遊戲式學習為依據，通過《魚蛋遊戲》的數位教學實驗，在香港一所小學的課堂中進行質性個案研究。研究結果顯示，學生普遍存在乘法結構的錯誤認知，概念理解與學業表現之間存在落差。而透過遊戲中的鷹架支援、即時回饋與同儕互動，學生能進一步調適其原有圖式，實現概念轉變。研究指出，數位工具在建構主義數學教育中具有重要價值，尤其在處理結構性錯誤概念方面。

【關鍵字】建構主義學習；鷹架支援；數位遊戲式學習；乘法結構

1. 前言

數位科技的發展使數位遊戲成為數學教學中的重要工具，特別在初小階段，學生對乘法結構（如乘數與被乘數）的理解經常出現混淆。建構主義理論強調學習者的主動參與、探索與反思，這些原則被逐漸融入遊戲化的設計中。本研究旨在探討建構主義導向的數位遊戲如何幫助香港小二學生克服乘法中的結構性誤解，並促進概念轉變。

2. 文獻回顧

建構主義與概念轉變與遊戲教學策略 建構主義學習理論主張，知識並非由教師單向灌輸，而是學習者在與環境互動中主動建構而成（Piaget，1962）。在 Piaget 的理論中，學習是一個圖式調整(adaptation of schemas)的過程。當新經驗能被納入既有圖式中，學習者會進行同化；但若舊有圖式無法解釋新情境，便會進入認知失衡，產生不安與困惑感，進而觸發學習動機。此時，學習者需透過調適，重建或修改原有圖式，以融合新經驗。當新的認知結構成功建立後，個體便會達致新的平衡狀態。圖式調整即念理解轉變的核心機制。Obikwelu 與 Read（2012）提出建構主義框架下的遊戲教學策略包括 1)示範，2)反思，3)，策略形成，4)鷹架探索，5)回顧和 6)表達。幫助學生在互動中建構知識。這些策略不僅強調學習過程中的錯誤修正，提供適度挑戰與鷹架，增加學習的自由度，並透過與同儕及教師的互動進行知識的社會建構。

數位遊戲式學習與數學教育 Gee（2003）指出數位遊戲式學習（DGBL）結合了視覺化、互動性與即時回饋，為現代「數位原住民」(Prensky，2005）提供了吸引且有效的學習方式。Reimer 和 Moyer（2005）同時指出，數位遊戲能將抽象的數學符號轉化為具體的視覺表徵，有助於學生理解如乘法、分數等需結構化思維的主題。特別是在乘法教學中，數位遊戲可以幫助學生從錯誤中反思，深化對乘法語意與結構的理解。

3. 研究方法

本研究採用質性個案研究法，研究工具為《魚蛋遊戲》，探索遊戲如何幫助學生理解乘法結構，區分乘數與被乘數。研究對象為香港一所小學的 21 名小二學生，年齡介於 7 至 9 歲，來自中低社經背景家庭，整體學業表現屬中等至偏低。《魚蛋遊戲》是一款建構主義導向的遊戲，學生需根據乘法算式（如「4×3」）正確排列魚蛋結構，系統以結構準確度而非總數作為評分標準，即時提供「正確／錯誤」回饋，幫助學生建構對乘法的深層理解。

課前訪談 研究員對五位學生進行個別訪談。受訪學生按學業表現選出，包括兩位成績優異、一位中等、兩位成績較低。訪談目的為探討學生對乘法結構的既有概念。每位學生被要求以口語說明及使用木塊實物操作，講解「2×3」與「3×2」的差異。

遊戲設計和施行 本研究依據建構主義教學原則設計的戲名為《魚蛋遊戲》。遊戲以學生扮演魚蛋小販的情境，根據所給的乘法算式準備魚蛋。遊戲畫面提供四枝竹籤，學生需點擊畫面，將魚蛋加入竹籤上，排列出正確的乘法結構。例如，當題目為「4 × 3」時，學生需在三枝竹籤上各放四顆魚蛋，共 12 顆；若錯誤地在 4 枝竹籤上各放 3 顆，雖總數仍為 12，但因結構不符，系統會判定為「錯誤」。遊戲以結構準確度而非總數作為評分標準，並提供正確或錯誤即時回饋。遊戲透過重複且具 Vygotsky（1978）鷹架支援的操作，強化學生對乘法中乘數與被乘數的角色理解。研究團隊設計了一節 30 分鐘的課堂，結合直接講授與《魚蛋遊戲》的操作。課堂分為以下步驟 1)示範與教學：教師透過例句與圖像說明乘法結構，並示範遊戲操作。2)遊戲操作：學生兩人一組進行遊戲，透過反覆操作與同儕互動修正錯誤。3)紙筆評估：課堂結束前，學生完成紙筆評估，測試其是否能理解概念。

課後訪談 研究團隊於課後選取四位學生進行深入訪談，其中兩位來自課前訪談組，課堂前他們誤解概念，但在遊戲中表現良好；另兩位則由教師觀察選出有著概念轉變的學生。

4. 研究結果與討論

學生混淆乘法結構 許多學生對乘法符號（如 2×3）與其語言或概念意義之間的對應關係存在混淆。於課前口頭評估中，學生對「2×3」與「3×2」的正確解釋產生明顯分歧。例如，當被問到如何表示「三個二」時，部分學生指應寫為「3×2」，並引發如「你錯了」等課堂爭辯。

概念理解與學業表現的落差 課前訪談結果顯示，學生對乘法結構的概念理解與其傳統考試成績之間缺乏明顯關聯。在五位受訪學生中，有兩位表現出明顯誤解，但其學業成績卻分屬兩極。舉例而言，小明為一名數學成績優異者，過往兩次考試皆獲 A 級，但他將「2×3」解釋為「3+3」；另一位低分學生小美則認為「2×3」與「3×2」完全相同，皆代表「3+3」。

遊戲促進概念轉變 《魚蛋遊戲》通過即時回饋與鷹架支援，有效引發學生的認知失衡，促使其調整既有圖式。例如，當學生因結構錯誤而收到「錯誤」提示時，會重新思考並修正操作。此外，同儕互動也在概念轉變中發揮了重要作用，學生透過討論共同建構了對乘法結構的更準確理解。

5. 結論

本研究顯示，嵌入建構主義原則的遊戲能有效促進小學生理解乘法結構，尤其釐清「乘數」與「被乘數」的語意與符號關係。透過即時回饋、重複操作與同儕合作，學生經歷認知調適，逐步建構正確概念。本研究建議將數位遊戲作為傳統教學的補充策略，未來可探討其遷移效果及於課堂中的長期應用。

參考文獻

Gee, J. P. (2003). *What video games have to teach us about learning and literacy.* Palgrave Macmillan.

Obikwelu, C., & Read, J. C. (2012). Children's learning in digital games: A conceptual framework. In P. Felicia (Ed.), *Proceedings of the 6th ECGBL (pp. 400–408)*. Academic Publishing.

Piaget, J. (1962). *Play, dreams and imitation in childhood.* Routledge & Kegan Paul.

Prensky, M. (2005). *Digital natives, digital immigrants. On the Horizon, 9(5), 1–6.*

Reimer, K., & Moyer, P. S. (2005). Third graders learn about fractions using virtual manipulatives: A classroom study. *Journal of Computers in Mathematics and Science Teaching, 24(1), 5–25*

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes.* Harvard University Press.

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Turning Math Concepts Visible and Sound

Jessica Tsz Shan SO, Yee Nok CHOW*

Lingnan University Alumni Association (HK) Primary School, Hong Kong SAR

sts@luaaps.edu.hk, cyn@luaaps.edu.hk

Abstract: *In the current educational environment, cross-curriculum is increasingly valued. The mathematics platform Polypad not only provides electronic teaching aids for primary school mathematics, but also incorporates musical elements into mathematical concepts - the area and perimeter of 2D-shapes, and even as numbers and number bars, can all be represented by pitch, rhythm and melody. Through perimeter teaching activities, our school allows students to interpret abstract mathematical concepts through music, providing students with a rich learning experience. This article records the teaching design of our school's mathematics classroom, allowing students to make abstract mathematical concepts visual and interesting through music; and in line with the purpose of "every child shine", a "Mathematics Orchestra" was formed to allow students who like music and mathematics to show more outstanding creativity.*

Keywords: mathematics, music, Polypad, cross-curricular, visualization

透過可見的音樂演繹抽象的數學世界

蘇梓珊，周苡諾*
嶺南大學香港同學會小學，香港
sts@luaaps.edu.hk，cyn@luaaps.edu.hk

【摘要】 在當前教育環境中，跨學科的合作越來越受到重視。數學平台 Polypad 提供了小學數學電子教具的同時，亦將數學概念融入了音樂元素——圖形的面積、周界，以至數類如數粒、數條等均，都可以以音高、節奏和旋律表示。本校透過周界教學活動，讓學生透過音樂演繹抽象的數學概念，為學生提供了豐富的學習體驗。本文記錄了本校由數學課堂的教學設計，讓學生把抽象的數學概念透過音樂變得可視化及有趣；以及本着「希望每位孩子也發亮」宗旨，組成「數學管弦樂隊」，讓對喜歡音樂及數學的學生，展現出更加出色的創造力。

【關鍵字】數學；音樂；Polypad；跨學科；可視化

1. Polypad 數學平台在數學及音樂科的應用

Polypad 所提供的形象化工具使課堂更加實用且可視化，提供了大部分小學數學課所需的教具，從而幫助學生將抽象的數學概念具象化，進一步提升學習效果。根據老師們的觀察，喜歡音樂及數學的學生，能夠利用數學工具發展潛能，展現出更加出色的創造力和解決問題的能力。

2. 本校的活動計劃實踐

2.1. 第一階段：全級四年級的 Polypad 活動

在試後活動中，全級四年級的學生參加 Polypad 活動，學習數學科周界的課題。具體步驟如下：

2.1.1. 介紹 Polypad 平台

先介紹圖形的周界，並向學生介紹 Polypad 的功能，如何使用該平台進行圖形的繪製和周界的計算。同時如何利用平台內含的音樂元素，配以不同的樂器製作可「聽得到」周界。

2.1.2. 圖形創作

讓學生在 Polypad 中選擇不同的圖形（如正方形、長方形、三角形等），利用不同圖形邊的不同數目，讓學生可以透過「聽得到的周界」旋律及節奏辨識圖形，學生亦利用不同圖形的周界製作樂章。

2.1.3. 作品提交

學生將他們的作品整理好，並上傳到 Google Classroom(圖 1)， 以便老師和同學進行評價和交流。

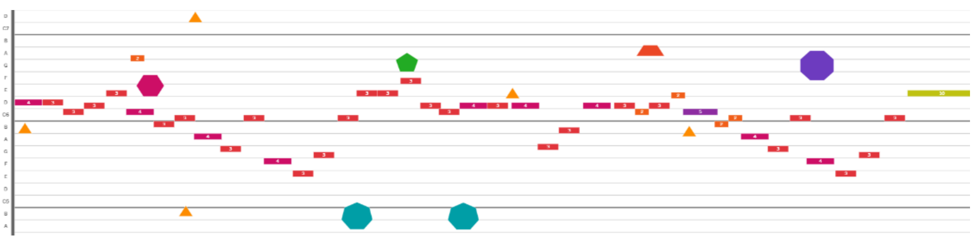


圖 1 學生在 Google classroom 遞交的作品

2.2. 第二階段：音樂創作與演奏——「數學管弦樂隊」

在第一階段活動的基礎上，老師們觀察到那些對音樂有較高能力並喜歡數學的學生，能夠利用這數學工具，展現出出色的創造力和解決問題的能力。所以我們發展至第二階段，進一步深化音樂與數學的結合而組成了數學管弦樂隊。

2.2.1. 「數學管弦樂隊」

邀請第一階段有興趣並表現出色的學生組成了「數學管弦樂隊」，成員根據興趣和特長進行分工。一部分學生專注於 Polypad 的音樂創作，另一部分則運用音樂知識運用鋼琴、小提琴和敲擊樂器進行音樂的實踐演奏。

最後兩組學生合奏演出，在開放日展示他們的成果。這不僅體現了數學與音樂的結合，也為學生提供了一個展示自己才華的舞台。

3. 總結

通過這兩個階段的活動，我們觀察到學生非常投入，而然數學活動變得更有趣及可視化。而在第一階段活動的基礎上學生已掌握如何利用不同的圖形如正方形、長方形、三角形等製作樂章(*Bloom's taxonomy 裡的 1. 記憶 2. 理解 3. 應用*) 。在 Bloom's taxonomy 裡中，當學生已有足夠的學習經驗，便會有能力進入第四至六層的高階思維。而在第二階段的過程中， 學生除了使用第一階段教授以圖形周界製作樂章，他們的作品裡更自行探索出利用分數、角度、百分數等的音樂演繹(圖 2)，展現了他們的創造力和解難能力。 (*Bloom's taxonomy 裡的高階思維部分: 4. 分析 5. 評估及 6. 創造。*)觀察到學生同時，這樣的跨學科學習方式增強學生對數學和音樂的興趣，學生在愉悅的氛圍中成長與學習。透過音樂藝術的引入，學生能夠欣賞數學的美，並從中獲得情感的共鳴。這種學習方式不僅提升了學生對數學的興趣，還幫助他們在視覺和聽覺上更好地理解數學知識。

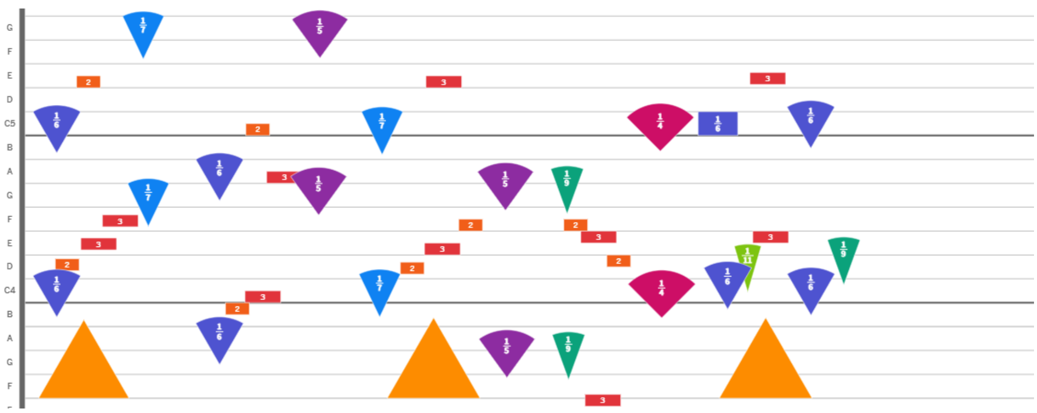


圖 2 數學管弦樂隊加入分數等元素的作品

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

Using Computational Thinking Education in Interdisciplinary Collaboration to Cultivate Lifelong Learning in Students to Meet Future Needs: A Case Study of the "Aqua Odyssey" Integrated Curriculum

Fong CHU[1*], Huihuan WU[2*]
[1,2]Caritas Fanling Chan Chun Ha Secondary School, Hong Kong SAR
candicechu126@gmail.com，kiss0765@yahoo.com

Abstract: *This study explores how to cultivate the core literacy of middle school students through an interdisciplinary integrated teaching model in the context of the Artificial Intelligence era. The viewpoint raised by Professor Lahani of Harvard Business School deeply reveals the importance of cultivating unique human capabilities in the AI era. In an exclusive interview, the Education University of Hong Kong FLASS interview of Professor Stephen Chiu mentioned the 4C core skills that are difficult to effectively cultivate in current generative AI - critical thinking, communication, teamwork and creativity. The study designed and implemented an integrated curriculum that combines science and computer subjects. By allowing students to transform scientific knowledge such as the physical changes of water and the water cycle into interactive applications and use computational thinking to solve practical problems. This study provides a practical reference for the innovative development of computational thinking education in the era of Artificial Intelligence.*

Keywords: 4C Skills; Computational Thinking; Interdisciplinary Teaching Model

Kong, S. C., Hsu, T. C., & Zhao, J. H. (Eds.). (2025). *Proceedings of the 9th International Conference on Computational Thinking and STEM Education*. Hong Kong: The Asia-Pacific Society for Computers in Education.

透過結合運算思維之跨學科學習培養學生終身學習以迎接未來需求

——以『水漫旅程』為案例

朱芳 [1*]，吳惠歡 [2*]
[1,2] 明愛粉嶺陳震夏中學，香港
candicechu126@gmail.com，kiss0765@yahoo.com

【摘要】 探討在人工智能時代背景下，透過跨學科教學模式培養中學生核心素養。哈佛商學院拉哈尼教授提出的觀點，深刻揭示了在 AI 時代培養人類獨特能力的重要性，香港教育大學趙永佳教授在專訪中提到當前生成式 AI 難以有效培養的 4C 核心技能－批判性思考（Critical Thinking）、溝通（Communication）、團隊合作（Collaboration）及創新（Creativity and Innovation）。研究設計並實施了一套融合科學科與電腦科的跨學科課程。讓學生將水的物態變化和水循環等科學知識轉化為可互動的應用程序，並運用運算思維解決實際問題。本研究為運算思維教育在人工智慧時代的創新發展提供了實踐參考。

【關鍵字】 4C 技能；運算思維；跨學科教學模式

1. 前言

隨著人工智慧技術的快速發展，教育領域面臨培養新型人才的迫切需求。本研究聚焦培養學生 4C 技能，將運算思維引入科學教育領域，透過"水循環與水的三態變化"這一典型科學主題，探索跨學科整合教學模式。運算思維的核心要素—問題分解、模式辨識、抽象化和演算法設計，為學生理解抽象科學現象提供了新的認知工具。跨學科教育中學生應用運算思維，將科學現象分解為可操作的模組，辨識其中的模式和規律，最終設計出能夠模擬水循環過程的演算法和程式。水循環過程因其系統性和可模型化的特點，成為展現運算思維應用價值的理想載體。

2. 案例教學設計背景

跨學科教學設計充分參考香港教育局頒布的《小學常識科課程指引》及《中學科學科課程指引》的相關要求。在科學學科部分，教學內容緊密對應香港課程中"物質世界"和"地球與環境"兩大學習範疇，重點涵蓋水的物態特徵及其轉換條件，特別是溫度變化對固態、液態和氣態相互轉化的影響機制，包括蒸發、凝結、降雨和徑流等自然過程，及整個水循環系統的動態平衡原理。在電腦學科部分，教學內容依據香港《運算思維-編程教育》補充文件指引，包括 MIT App Inventor 可視化編程工具的基礎應用，著重培養學生運用運算思維四大核心要素（分解、模式識別、抽象化和技術設計）解決問題的能力。透過跨學科合作教學模式，學生能夠將科學概念轉化為可執行的程序邏輯，實現理論知識與實踐能力的有機結合，落實香港教育局提倡的 21 世紀能力教育目標。

3. 案例教學概述

3.1. 教學重點與困難分析

教學重點主要體現在三個方面：首先，學生需掌握將科學概念轉化為程序邏輯的能力，特別是如何運用溫度變量控制水的物態顯示；其次，重點培養算法設計能力，透過順序結構和循環控製完整實現水循環過程的程序模擬；最後，要確保交互功能設計與科學原理保持高

度一致，使程序操作能準確反映自然規律。在教學困難方面，首要挑戰在於如何將抽象的蒸發速率等科學現象轉化為可量化的程式參數；其次，需要解決將連續的自然過程分解為離散程序步驟的技術難題；具挑戰性的是要求學生在程式設計過程中持續驗證程式邏輯是否符合科學原理，建立雙向校驗機制。這些難點特別考驗學生的抽象思考能力和跨學科知識整合能力。

3.2. 教學策略分析

多元化的教學策略體系促進學生的學習成效。學生著手開發"水漫旅程"應用程式，以"模擬水循環系統"為驅動性問題，引導學生透過完整的專案開發流程整合跨學科知識。在教學支援方面，根據學生不同的學習需求提供分層指導，包括示範案例、步驟指引和個別化回饋等支援手段。最後，在評估方式上，採用形成性評量與過程性評量結合的綜合評量體系，既關注最終作品質量，也重視學生在計畫開發各階段的表現與進步，透過多元化的評量指標全面衡量學習成效。

4. 案例教學設計

4.1. 界定問題（科學科實現）

在科學科的實驗觀察到的物態轉變現象，確保關鍵溫度點能準確呈現轉變過程；同時需要解決介面互動設計問題，包括溫度調節控制元件、內容展示面板等元素的佈局與功能實現。對於教授水循環知識，選擇典型氣象紀錄片中的水循環片段，將其分解為蒸發、凝結、降水和徑流四個關鍵環節；建立方位互動設計。這種方法能直觀展示水循環的動態連續性特徵。

4.2. 項目實踐與測試（電腦科實現）

在電腦實作課上，學生們分組進行"水漫旅程"應用程式的開發任務。教師首先透過生活實例（如電燈開關）來解釋事件驅動程式設計的概念，引導學生瞭解使用者操作與程式回應之間的關係。學生依照任務清單逐步完成專案：先建立包含主介面、三態展示和水循環模擬的多螢幕架構；加入冰、水、蒸氣等圖片素材，設定標籤元件顯示科學解釋文字；然後重點設計溫度調節滑桿，透過變數命名和條件判斷實現物態自動切換功能。在水循環模組開發中，學生需要編寫自動循環邏輯，使水循環中四個環節能夠順序執行，同時為每個環節添加文字轉語音的科普講解。測試環節中，各組透過模擬使用者操作驗證功能完整性，並使用偵錯工具修正發現的邏輯錯誤。整個實作過程強調科學準確性與程序邏輯的有機結合。

5. 案例教學建議

在小組合作環節，學生始終以問題解決和分析為導向，系統性地進行專案分解、方案設計、修改完善及成果分享等工作流程。在此過程中，學業表現優異的學生能夠突破傳統學習模式的限制，充分發揮個人潛能；而學習進度相對滯後的學生，則在教師的差異化指導及組員間的互助支持下，根據分工安排發揮自身優勢，有效履行團隊成員的職責。這種教學模式不僅能有效激發學生的求知慾和學習興趣，更能促進其主動參與知識建構，最終培養形成跨領域的綜合思考能力。

香港教育大學 The Education University of Hong Kong

Artificial Intelligence and Digital Competency Education Centre 人工智能及數碼能力教育中心

Co-organiser:

2025 CTE-STEM

Website URL:

https://www.eduhk.hk/ctestem2025/

Email:

ctestem2025@eduhk.hk

Supporting Organisations:

CoolThink@JC 賽馬會運算思維教育

Education Bureau
The Government of the
Hong Kong Special Administrative Region
of the People's Republic of China

AiTLE

香港電腦教育學會 The Hong Kong Association for Computer Education